

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problems Mailbox.**

PCT

世界知的所有権機関  
国際事務局

特許協力条約に基づいて公開された国際出願



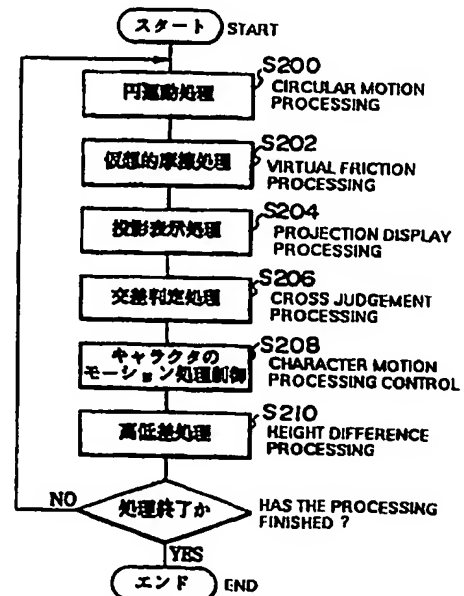
(51) 国際特許分類6 A63F 9/22, G06T 15/70		A1	(11) 国際公開番号 WO97/46295
			(43) 国際公開日 1997年12月11日(11.12.97)
(21) 国際出願番号 PCT/JP97/01896		(22) 国際出願日 1997年6月4日(04.06.97)	
(30) 優先権データ 特願平8/165224 1996年6月5日(05.06.96) 特願平8/186666 1996年6月27日(27.06.96)		JP JP	(74) 代理人 弁理士 稲葉良幸, 外(INABA, Yoshiyuki et al.) 〒105 東京都港区虎ノ門3丁目5番1号 37森ビル803号室 TMI総合法律事務所 Tokyo, (JP)
(71) 出願人 (米国を除くすべての指定国について) 株式会社 セガ・エンタープライゼス (SEGA ENTERPRISES, LTD.)[JP/JP] 〒144 東京都大田区羽田1丁目2番12号 Tokyo, (JP)		(81) 指定国 AU, BR, CN, JP, KR, US, 欧州特許 (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).	
(72) 発明者; および (75) 発明者/出願人 (米国についてのみ) 加来徹也(KAKU, Tetsuya)[JP/JP] 峰裕一朗(MINE, Yuichiro)[JP/JP] 小野孝志(ONO, Takashi)[JP/JP] 芳賀憲夫(HAGA, Norio)[JP/JP] 大崎 誠(OSAKI, Makoto)[JP/JP] 山口貴之(YAMAGUCHI, Takayuki)[JP/JP] 関根紀裕(SEKINE, Norihiro)[JP/JP] 油井亮弥(YUI, Ryoya)[JP/JP] 西川さおり(NISHIKAWA, Saori)[JP/JP] 杉本哲也(SUGIMOTO, Tetsuya)[JP/JP] 吉田 茂(YOSHIDA, Shigeru)[JP/JP]		添付公開書類 国際調査報告書 補正書	

(54) Title: IMAGE PROCESSOR FOR GAMES

(54) 発明の名称 ゲーム用画像処理

(57) Abstract

In this image processor for games, a predetermined number of models (characters) are set in a virtual space, and these models are controlled so that they are moved in a predetermined direction in the virtual space, an image in this virtual space based on virtual view points being shown on a display means. This apparatus is provided with an image processing means, as one constituent part, for applying a virtual centripetal force to the models for the purpose of more realistically displaying the movements of the models. This processor has, in order to more realistically display the movements of the models and heighten a stage effect, and afterimage presentation processing means for presenting the movement paths as an afterimage and this processing means has a means for storing motion data as they are which were collected before the data on the actual motion of the models, and a display control means for displaying the stored data with the actual motion data.



# (57) 要約

このゲーム用画像処理装置は、仮想空間内に所定数のモデル（キャラクタなど）を設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、仮想視点に基づくこの仮想空間の映像を表示手段において表示するようにした装置である。この装置は、仮想空間に設定されたモデルの動きをよりリアルに表示できるようにするため、その1つの構成として、モデルに仮想的な向心力を与える画像処理手段を備える。また、この装置は、モデルの動きをよりリアルに表示しかつ演出効果を上げるため、その1つの構成として、モデルの移動軌跡を残像として表現するための残像表現処理手段を備え、この処理手段はモデルの現在モーション以前のモーションデータをそのまま記憶する記憶手段と、この記憶データをモデルの現在のモーションデータとともに表示する表示制御手段とを備える。

## 参考情報

PCTに基づいて公開される国際出願のパンフレット第1頁に記載されたPCT加盟国を特定するために使用されるコード

AL	アルバニア	ES	スペイン	LR	リベリア	SG	シンガポール
AM	アルメニア	FI	フィンランド	LS	レソト	SI	スロヴェニア
AT	オーストリア	FR	フランス	LT	リトアニア	SK	スロヴァキア共和国
AU	オーストラリア	GA	ガボン	LU	ルクセンブルグ	SL	シエラレオネ
AZ	アゼルバイジャン	GB	英国	LV	ラトヴィア	SN	セネガル
BA	ボスニア・ヘルツェゴビナ	GE	グルジア	MC	モナコ	SZ	スワジランド
BB	バルバドス	GH	ガーナ	MD	モルドヴァ共和国	TD	チャード
BE	ベルギー	GM	ガンビア	MG	マダガスカル	TG	トーゴ
BF	ブルキナ・ファソ	GN	ギニア	MK	マケドニア旧ユーゴスラヴィア共和国	TJ	タジキスタン
BG	ブルガリア	GR	ギリシャ			TM	トルクメニスタン
BJ	ベナン	HU	ハンガリー	ML	マリ	TR	トルコ
BR	ブラジル	ID	インドネシア	MN	モンゴル	TT	トリニダード・トバゴ
BY	ベラルーシ	IE	アイルランド	MR	モーリタニア	UA	ウクライナ
CA	カナダ	IL	イスラエル	MW	マラウイ	UG	ウガンダ
CF	中央アフリカ共和国	IS	アイスランド	MX	メキシコ	US	米国
CG	コンゴ	IT	イタリア	NE	ニジェール	UZ	ウズベキスタン
CH	スイス	JP	日本	NL	オランダ	VN	ヴェトナム
CI	コート・ジボアール	KE	ケニア	NO	ノルウェー	YU	ユーゴスラビア
CM	カメルーン	KG	キルギスタン	NZ	ニュー・ジーランド	ZW	ジンバブエ
CN	中国	KR	朝鮮民主主義人民共和国	PL	ポーランド		
CU	キューバ	KZ	大韓民国	PT	ポルトガル		
CZ	チェコ共和国	LC	カザフスタン	RO	ルーマニア		
DE	ドイツ	LI	セントルシア	RU	ロシア連邦		
DK	デンマーク	LK	リヒテンシュタイン	SD	スーダン		
EE	エストニア		スリランカ	SE	スウェーデン		

## 明 細 書

## ゲーム用画像処理

## 5 技術分野

本発明は、仮想的に設定したゲーム空間（以下、「仮想空間」と云う。）における画像（以下、「仮想画像」と云う。）の処理に係わり、特に、仮想空間内に設定された仮想なモデル（例えば、キャラクタと称される）の画面上での動きをよりリアルに表現できるゲーム用画像処理に関する。この画像処理の手法は、特に、3Dビデオゲーム機に  
10 好適である。

## 背景技術

コンピュータ・グラフィックス（CG）技術の発達により、仮想的に設定した仮想空間（「仮想世界」とも云う。）を立体的にリアルタイムに表現できるようになった。近年開発されている高速演算が可能な中央処理装置（CPU）、ビデオ表示プロセッサ（VDP）を搭載し、このコンピュータ・グラフィックス技術を高速にかつ経済的に利用しているのがビデオゲーム機の技術分野である。  
15

このビデオゲーム機では、ユーザ（遊戯者或いはプレーヤ、と云っても良い。）の操作内容に基づいてゲームプレイの内容を刻々と変化させるため、仮想空間内において被表示物体を高速で自在に動かす必要がある。このため、通常は、被表示体としてのモデル（例えば、キャラクタ（人物））を三角形或いは四角形のポリゴンと云う多角形の小片の集合から構成し、各ポリゴンの空間位置を同時に変化させながらこれらのモデルの動きを表現している。  
20

また、キャラクタの腕や脚等、被表示物体の特定の部分や面が同一の動きをする場合には、複数のポリゴンを集めたポリゴンデータ群を単位とし、ポリゴンデータ群ごとに空間位置を与え、この特定の部分や面を同時に動かしていた。  
25

近年、キャラクタをポリゴンによって構成し、仮想空間内のキャラクタの動きを所定



の仮想視点から捉えた映像をモニタに表現するようにした、いわゆる 3D (3次元) ゲームが市場において注目を集め、特に、複数の闘士の格闘をシュミレートしたものが高い人気を得ている (例えば、株式会社セガ・エンタープライゼス製の「バーチャファイター」(商標))。

5 この格闘シュミレーションゲームでは、遊戯者はコントローラに付いているスティックやパッド或いは釦を素早く操作して、画面に表示されている闘士が、スティック等の操作によって決められたコマンドに応じて動作するようにされる。闘士の動きは「モーション」と称されており、このモーションを実現するためのデータは、モーションキャプチャーの技術を用いて取得される。そして、必要に応じてこのデータを加工し、最終  
10 的なモーションデータとして 3D ビデオゲーム機において利用されている。

この種のビデオゲームにおいては、その商品価値を高めるために、キャラクタの動きをよりリアリティを持って表現することが望まれる。すなわち、例えば、実際の格闘の動作に極力近づけながら、より多彩な動きを付加することである。しかしながら、予想されるキャラクタの動きは極めて多岐にわたるために、このような目的を達成する上で  
15 未だ改善されなければならないことが数多くあった。もっとも、望まれるモーションデータを予め全て作成してメモリに記憶させたり、このようなモーションを得るための特性式を得ること等も考えられるが、これではデータ量が膨大になり、リアルタイムの処理が行えない問題がある。

したがって、本発明の主目的は、画像処理に要する演算量やデータ量を極力少なく抑  
20 えて、よりリアルタイムに、かつ、キャラクタの画面上の動きをよりリアルに表現できるゲーム用画像処理の手法を提供することである。

この主目的の観点から、従来ビデオゲーム機が有していた問題の具体的な第 1 の側面は次のようなものである。

A 1. 3D ビデオゲームでは、2次元の画面に仮想視点からの映像が投影変換されて表現されるために、画面の奥行き方向に (仮想空間の z 方向)、遊戯者が望むように  
25 闘士を移動させることは難しく、したがって、遊戯者が一つの闘士を他の闘士の周りを周回させるようにすることは何等配慮されていなかった。そこで、このようなキャラク

タの周囲の動きを改善することを、本発明の1つの具体的な目的とする。

A 2. 従来 of 3 D ビデオゲーム機は、仮想視点からの映像を表示するようにしていたため、闘士を遮るような位置に、壁等の構造物が仮想空間内に設定されていた場合、キャラクタが遮られて表示されてしまうと云う問題があった。そこで、このようにキャラクタが構造物に遮られるという事態を改善した表示を行うことを、本発明の別の具体的な目的とする。

A 3. 従来 of ビデオゲーム機では、キャラクタのモーションを例えばスプライン関数によって順次作って行く方法、又は予め定められたパターンを順次フレームに再現して行く方法が採用されている。しかしながら、従来 of ビデオゲーム機ではこのモーションが固定されており、相手側キャラクタの動き等に合わせてモーションを補正することができなかった。そこで、このようなモーション補正を行えるようにすることを、本発明のさらに別の具体的な目的とする。

さらに、上記主目的の観点から派生することとして、その商品価値を高めるために、上述した第1の側面に加えて、キャラクタの動きなどに演出効果を高めた画面表示を行うことが望まれている。この要望に鑑みた従来 of ビデオゲーム機の問題の第2の側面は、具体的には以下のように説明できる。

B 1. C G 画像の演出効果を高める技術として、モーション・ブラーが知られている。このモーション・ブラーによれば、一つの画素に多くのレイを発生させ、その平均をとった色を付けることによってピンボケや動きのある絵を作ることができる。

一方、近年では、ビデオゲーム等のC G 映像の分野では、さらに演出効果を高めるために、キャラクタの動きに人間の視覚生理上ありがちな残像と一緒に表示することが行われている。例えば、闘士が振り回す刀の軌跡に残像を付すような場合である。そこで、当業者は、残像としてのポリゴンを闘士のモーションに合わせて計算して、この残像を闘士の近くに表示するようにしていた。

しかしながら、予想されるキャラクタの動きは極めて多岐にわたるために、全ての場合に合わせて、多くの形態の残像用のポリゴンを作成してこれをメモリに記憶させることは、限られたコンピュータグラフィックス装置の能力を制限することになるばかりか、

キャラクタのモーションに合わせて残像のポリゴンを計算することは、コンピュータグラフィックス装置に対する負荷が大きい。そこで、CG画像を形成する上で、演出効果を高めることの配慮があっても大きな計算負荷を与えることが無く（より詳しくは、このような負荷を少なくして）、残像映像を実像画面と同時に表示することができるようにすることを、本発明のさらに別の具体的な目的とする。

B 2. 従来の画像処理装置を利用したゲーム装置において、画面に砂や水飛沫等の飛翔体を表示することが行われている（例えば、（株）セガ・エンタープライゼス製「セガラリー」（商標））。しかしながら、このような水飛沫や砂飛沫は、テクスチャーをポリゴンにマッピングしていただけであったため、モデル（車等）の動きを正確に飛翔体  
10 体に反映させることができなかった。そこで、モデルの動きをより正確に飛翔体に反映させるようにすることを、本発明のさらに別の具体的な目的とする。

B 3. 従来の画像処理装置では、仮想空間内を落下するモデルの運動をシミュレートする上での品質は十分でなかった。そこで、仮想空間内を自由落下する運動のシミュレートをより高品質なものにすることを、本発明のさらに別の具体的な目的とする。

15 B 4. 従来の画像処理装置においては、仮想空間内の区画と移動するモデルとの当  
たりを判定し、この判定が肯定された場合には、モデルが区画を越えることがないよう  
にその動きを制限していた。通常、この種の区画としては、四角や円形等の定型タイプ  
のものが設定されていた。この場合は、区画が一般的な形状であるために、キャラクタ  
を区画から追い出す運動の計算をすることも容易であった。しかしながら、この区画が  
20 不定形の場合には、従来のコンピュータグラフィックス装置では、このことが容易に対  
応できないという問題があった。そこで、不定形の区画であっても、区画とモデルとの  
衝突判定肯定後のモデルの動き（モーション）の計算処理が的確且つ容易に実行できる  
ようにすることを、本発明のさらに別の具体的な目的とする。

25 B 5. 従来、ゲーム装置に適用される画像処理の分野では、周期的な絵の列（例えば、繰り返す波の絵、縄跳び等）を表現する場合、これらの絵の列を手作業で作成してテクスチャー列とし、これを順次ポリゴンにマッピングすることを繰り返すことにより、同じ動作を繰り返す一連の映像を表現していた。

このようなテクスチャーの列を作成するのは、多くの手作業を要することになるので、アプリケーションソフトウェアを利用することが試みられている。このようなアプリケーションソフトウェアとしては、例えば、商品名：

Alias/Wavefront (Alias/Wavefront 社 (110 Richmond St. East Toronto, Ontario Canada  
5 M5c 1p1) 製) が知られている。このソフトウェアを利用する場合、所定のパラメータをソフトウェアに与えれば目的とするテクスチャー列を得ることができる。

しかしながら、従来はこの種のソフトウェアを利用した場合、テクスチャー列の最初と最後、すなわち、テクスチャーの柄(形態)が遊戯者にとって連続しないような絵になることが通例であった。

10 したがって、従来 of 画像処理の分野では、繰り返す絵の列を作る上で未だ改良する余地があった。そこで、画像処理によりこの種のテクスチャー列を作成できるようにすることを、本発明のさらに別の具体的な目的とする。

B 6. 従来 of 画像処理装置では、再生画像の演出感を高めるために、キャラクタの動作の再生速度を遅くしてスロー再生を行う場面を有していた。しかしながら、このス  
15 ロー再生の演出はその効果が限られていた。そこで、より効果的な演出感を醸し出す上での再生速度の変化を可能にしたゲーム用画像処理の手法を提供することを、本発明のさらに別の具体的な目的とする。

#### 発明の開示

20 本発明によって、その主目的である、画像処理に要する演算量やデータ量を極力少なく抑えて、よりリアルタイムに、かつ、キャラクタの画面上の動きをよりリアルに表現できるゲーム用画像処理の手法が様々な観点から提供される。本発明では、この主目的から派生する種々の具体的な目的を達成するため、以下のような種々の構成が採られる。

本発明の 1 つの側面を成す一連の構成は以下のものである。すなわち、本発明は、  
25 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、仮想視点に基づくこの仮想空間の映像を表示手段において表示するようにしたゲーム用画像処理装置であって、前記モデルに仮想的な向心力を

与える画像処理手段を備えることを特徴とする。また、このゲーム用画像処理装置を備えたゲーム装置が提供される。さらに、この画像処理手段を実行する一連の手順を含む情報を記録した記録媒体が提供される。

さらに、本発明は、この種のゲーム用画像処理装置において、前記モデルが移動して  
5 いる時と静止している時との摩擦量を変えて、このモデルに与える画像処理手段を備えることを特徴とする。

またさらに、本発明は、この種のゲーム用画像処理装置において、前記モデルが置かれるステージの表面形状に合わせて前記モデルの投影像を表示させる画像処理手段を備えることを特徴とする。

10 さらに発明は、この種のゲーム用画像処理装置において、前記仮想視点が表示対象としての前記モデルに対して成す視野領域と、表示対象として設定されていない他のモデルとの重なり判定を行い、この判定が肯定された場合に、この他のモデルが表示されないような画像処理手段を備えることを特徴とする。

さらに本発明は、この種のゲーム用画像処理装置において、前記モデルの予定された  
15 移動軌跡の終了点に対して、この終了点とは異なる目標点が設定され、前記終了点をこの目標点に一致させるように移動軌跡が補間される画像処理手段を備えることを特徴とする。

さらに本発明は、この種のゲーム用画像処理装置において、前記モデルが前記仮想空間内に置かれている高低差を求め、この高低差に応じてこのモデルに対して与えられて  
20 いる動作を補正する画像処理手段を備えることを特徴とする。

このように構成することにより、キャラクタ等仮想空間に設定されたモデルの動きをより現実感豊かに表示することができる。具体的には、一つのモデルが他のモデルの周りを周回する画像が容易に生成される。さらに、モデルが置かれているステージの状態をモデルの動きに反映させることができる。さらに、ステージの凹凸をモデルの投影像  
25 に正確かつ簡単に反映させることができる。さらに、モデルを遮るような位置に、壁等の構造物が仮想空間内に設定されていた場合でもこのモデルが遮られることなく表示される。また、モデルのモーションを相手側のモデルの動き等に合わせて補正すること

ができるので、モデルの多彩な動きを確実に表現することができる。さらに、モデルの高低差を考慮しているために、より現実感に優れた映像を生成することができる。

また、本発明の別の側面を成す一連の構成は以下のように開示される。その1つの具体的構成として、本発明は、仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、所定位置の視点に基づくこの仮想空間の映像を表示手段に表示するようにしたゲーム用画像処理装置であって、モデルの移動軌跡を残像として表現するための残像表現処理手段を備え、この処理手段は前記モデルの現在モーション以前のモーションデータをそのまま記憶する記憶手段と、この記憶データを前記モデルの現在のモーションデータとともに表示する表示制御手段とを備えたことを特徴とする。この表示制御手段は、前記記憶手段のデータに演出を加えて表示するための演出手段を備えることができる。この演出手段は、半透明の画像処理を行うことができる。

別の具体的構成として、本発明は、仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、所定位置の視点に基づくこの仮想空間の映像を表示手段に表示するようにしたゲーム用画像処理装置であって、前記モデルの移動特性を算出する算出手段と、前記仮想空間内に飛翔体を存在させるとともに、この飛翔体の一連運動を前記算出手段による算出結果に応じて制御する飛翔体制御手段とを備えることを特徴とする。さらに別の具体的構成として、本発明は、仮想空間内に設定されたモデルの自由落下運動をシミュレートするゲーム用画像処理装置において、前記モデルの全体運動をシミュレートする第1の手段と、前記モデルに周回運動を付与する第2の手段とを備えることを特徴とする。

さらに別の具体的構成として、本発明は、仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、所定位置の視点に基づくこの仮想空間の映像を表示手段に表示するようにしたゲーム用画像処理装置において、不定形の区画を前記仮想空間内に設定する区画設定手段と、この区画とこのモデルとの間に所定のベクトルを設定する手段であり、この区画を越えてこのモデルが移動しようとする場合には、このベクトルをモデルが区画を越えない方向に設定

するベクトル設定手段と、このベクトルに応じて前記モデルの移動を制御するモデル移動制御手段と、を備えることを特徴とする。

さらに別の具体的構成として、本発明は、所定のパラメータを与えて周期的に変化するテクスチャーを形成する装置において、第1のテクスチャー列を作る第1の手段と、  
5 第2のテクスチャー列を作る手段とを備えるとともに、第1のテクスチャー列の最後或いは最初のテクスチャーと、第2のテクスチャー列の最初或いは最後のテクスチャーとが連続する形態になるようにそれぞれの手段にパラメータを与えるとともに、第1のテクスチャー列と第2のテクスチャー列の少なくとも一方の透明度を漸増或いは漸減して両方のテクスチャー列の最初のテクスチャー同士から順に重ねて目的とするテク  
10 スチャー列を形成することを特徴とする。

また本発明は、周期的に変化するテクスチャーを形成する方法において、所定の方向に変化する第1のテクスチャー列を作る第1の工程と、同様に所定方向に変化する第2のテクスチャー列を作る工程とを備えるとともに、第1のテクスチャー列の最後或いは最初のテクスチャーと、第2のテクスチャー列の最初或いは最後のテクスチャーとが連  
15 続する形態になるようにするとともに、第1のテクスチャー列と第2のテクスチャー列の少なくとも一方の透明度を漸増或いは漸減して両方のテクスチャー列の最初のテクスチャー同士から順に重ねて目的とするテクスチャー列を形成することを特徴とする。

さらに別の具体的構成として、本発明は、仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、所定位置  
20 の視点に基づくこの仮想空間の映像を表示手段に表示するようにしたゲーム用画像処理装置において、二つのモデル間に所定の条件が成立したか否かを判定する第1の手段と、前記条件が成立したことをそれぞれのモデルの運動に反映させる第2の手段と、この条件が成立したときに一方のモデルの運動の再生速度を他方のモデルに対して変化させる第3の手段とを備えることを特徴とする。

25 このように構成することにより、仮想空間でのモデルの動きをよりリアルに表現できるとともに、演出感豊かなゲーム画像を生成することができる。より詳細には、その1つの具体的構成によれば、ゲーム画像を形成する上で、演出効果を高めることの配慮が

あっても大きな計算負荷を与えることが無いゲーム用画像処理装置を提供することができる。また別の具体的構成によれば、このような負荷を少なくして、残像映像を実像画面と同時に表示することができるゲーム用画像処理装置を提供することができる。さらに別の具体的構成によれば、モデルの動きを正確に飛翔体に反映させることができる

5 ゲーム用画像処理装置を提供することができる。さらに別の具体的構成によれば、仮想空間内を自由落下する運動のシミュレートをより高品質なものにできるゲーム用画像処理装置を提供することができる。さらに別の具体的構成によれば、不定形の区画であっても、区画とモデルとの衝突判定肯定後のモデルの動き（モーション）の計算処理が的確且つ容易に実行できるゲーム用画像処理装置を提供することができる。さらに別の

10 具体的構成によれば、画像処理を利用しても、繰り返して表示しても違和感無く連続映像が再現できる一連のテクスチャ列を作成することができる。さらにまた、別の具体的構成によれば、より効果的な演出感を醸し出す上での再生速度の変化を可能にしたゲーム用画像処理装置を提供することができる。

15 図面の簡単な説明

図1は本発明の実施例に係る画像処理装置としてのビデオゲーム機のブロック図である。

図2は第1の実施例におけるメインの処理を示すフローチャートである。

図3は円運動処理におけるキャラクタの円運動を示す斜視図である。

20 図4は円運動処理の詳細フローチャートである。

図5は仮想空間の上から円運動を示した模式図である。

図6は円運動の終了状態に相当する斜視図である。

図7は仮想的摩擦処理の原理図である。

図8は仮想的摩擦処理の詳細フローチャートである。

25 図9は投影表示処理の原理図である。

図10は交差判定処理の原理図である。

図11は交差判定処理の詳細フローチャートである。



図 1 2 は内接円て近似された構造物の  $x-z$  平面図である。

図 1 3 は交差判定処理の具体例を示す原理図である。

図 1 4 は交差判定処理の他の具体例を示す原理図である。

図 1 5 は壁状の構造物を近似した態様を示す原理図である。

5 図 1 6 はキャラクタのモーションの標準態様を示した斜視図である。

図 1 7 はキャラクタモーションの補間された態様を示す斜視図である。

図 1 8 はキャラクタのスプライン関数によって移動する軌跡を示す概念図である。

図 1 9 はキャラクタのモーション処理制御動作を示すフローチャートである。 図 2  
0 は二つのキャラクタの間に高低差がある状態を示す模式図である。

10 図 2 1 はキャラクタの高低差処理を示すフローチャートである。

図 2 2 はこの処理の時間流れを説明するための流れ図である。

図 2 3 は高低差処理がされた状態を示す模式図である。

図 2 4 は人物類似のキャラクタの一例の正面を示す、写真である。

図 2 5 はその第 2 の例を示す、写真である。

15 図 2 6 はコントロールパッドが拡大された正面図である。

図 2 7 は混合技のファイル構成を示す図である。

図 2 8 は混合技を展開するためのフローチャートである。

図 2 9 は混合技処理過程で表示される画面を表示した画面正面図である。

図 3 0 は同処理過程で表示される他の画面を表示した正面図である。

20 図 3 1 は同処理過程で表示される他の画面を表示した正面図である。

図 3 2 は同処理過程で表示される他の画面を表示した正面図である。

図 3 3 は同処理過程で表示される他の画面を表示した正面図である。

図 3 4 は第 2 の実施例におけるメインの処理を示すフローチャートである。

図 3 5 は残像処理がされたモデルの正面図である。

25 図 3 6 は残像処理の原理図である。

図 3 7 は残像処理の概略フローチャートである。

図 3 8 はその詳細フローチャートである。

図39は図37のフローチャートの詳細フローチャートである。

図40は跳ね上げられた飛翔体の軌跡を示す図である。

図41は飛翔体の着地発生を説明する図である。

図42は飛翔体の着地発生を説明する図である。

5 図43は飛翔体の着地発生を説明する図である。

図44は飛翔体の着地発生を説明する図である。

図45は跳ね上げ発生の説明図である。

図46は跳ね上げ発生の際のベクトル線図である。

図47は跳ね上げ発生の際のベクトル線図である。

10 図48は飛翔体運動処理のフローチャートである。

図49は飛翔体運動処理が適用されたキャラクタの背面図である。

図50はこの処理の概略フローチャートである。

図51はその詳細フローチャートである。

図52はその詳細フローチャートである。

15 図53はその詳細フローチャートである。

図54は飛翔体の運動形態を示す図である。

図55は飛翔体の運動形態を示す図である。

図56は自由落下運動処理がされた対象物の移動軌跡を示す図である。

図57は不定形区画とモデルとの衝突判定処理の原理図である。

20 図58はそのフローチャートである。

図59はテクスチャー列の作成原理図である。

図60はモデルによって飛翔体の跳ね上げ及び着地発生が生じた状態を示す正面図である。

図61はスロー再生処理の原理を説明するためのキャラクタの動作を説明するための側面図である。  
25

図62は図61を上方から見た平面図である。

図63はスロー再生状態にあるキャラクタの動作を説明するための側面図である。

図64は図63を上方から見た平面図である。

図65は本来の位置に復帰した状態のキャラクタを示す側面図である。

図66はこの処理の動作フローチャートである。

図67は不定形区画とモデルとの衝突判定処理によって表現されるモデルのモーション図である。

図68は図54の飛翔体運動処理における葉が風によって葉の高さ毎に影響を受ける係数を説明する図である。

発明を実施するための最良の形態

以下に、本発明の好適な実施例を、画像処理装置としてのビデオゲーム機について図面を参照して説明する。

#### 第1の実施例

ビデオゲーム機の第1の実施例を図1～図33に基づき説明する。

#### ハードウェアの説明

図1に、このビデオゲーム機のブロック図を示す。このビデオゲーム機によって、後述の各画像生成・処理1乃至6がそれぞれ実行される。

このビデオゲーム機は、装置全体の制御を行うCPUブロック10、ゲーム画面の表示制御を行うビデオブロック11、効果音などを生成するサウンドブロック12、CD-ROMの読み出しを行うサブシステム13などによって構成される。

CPUブロック10は、SCU(System Control Unit)100、メインCPU101、RAM102、ROM103、サブCPU104、CPUバス105等により構成されている。このブロックが本発明の画像生成装置の主体を担う。

メインCPU101は、内部にDSP(Digital Signal Processor)を備え、コンピュータプログラムを高速に実行できる。RAM102は、CD-ROMを読み出すサブシステム13から転送された各種ポリゴンデータを記憶する他に、メインCPU101のワークエリアとしても使用される。

ROM 103には、装置の初期状態において行う初期処理のためのイニシャルプログラムが格納される。SCU 100は、バス105、106、107を介して行われるデータの転送を統括する。また、SCU 100は、内部にDMAコントローラを備え、ゲームの実行中に必要な画像データをビデオブロック11内のVRAMへ転送する。

5     コントロールパッド2bは、ユーザの情報入力手段として機能し、操作に必要な各種ボタンが備えられている。サブCPU 104は、SMPC (System Manager & Peripheral Controller)と呼ばれ、メインCPU 101からの要求に応じ、コントローラ2bからペリフェラルデータを収集する機能を備えている。

10     メインCPU 101は、サブCPU 104から転送された周辺データに基づいて、ディスプレイに表示する画像の移動等の処理を行う。サブCPU 104は、コネクタ2a (本体側端子)に接続した周辺機器の種類を判定し、判定した周辺機器の種類に応じた通信方式に従って、周辺データを収集する。

15     ビデオブロック11は、図形生成手段として動作し、ポリゴンにより表示する画像を生成するためのVDP (Video Display Processor) 120と、背景画のための画像合成、陰面処理、クリッピングを行うVDP 130を備える。VDP 121及びフレームバッファ122、123へ接続される。

20     ディスプレイへ表示する仮想空間の画像を生成する際、表示に必要なポリゴンデータがメインCPU 101からSCU 100を介してVDP 120へ転送され、VRAM 121に書き込まれる。VRAM 121に書き込まれたポリゴンデータは、1画素あたり16ビット又は8ビットの色情報を含む描画用データとして、描画用のフレームバッファ122又は123へ格納される。格納された描画用データは、VDP 130へ転送される。メインCPU 101は、SCU 100を介して、描画を制御する制御情報をVDP 130へ供給する。VDP 130は、この制御情報に従って描画用データを処理する。

25     VDP 130はVRAM 131に接続されており、表示画面全体を上下左右に移動させたり、回転させたりするスクロール機能と、ポリゴンの表示順序を決定するプライオリティ機能 (Zソート或いはZバッファ) を備える。VDP 130は、描画用データをメモリ132を介してエンコーダ160へ出力する。エンコーダ160に出力された描

画用データは、ビデオ信号のフォーマットに変換された後、D/A変換されてモニタ装置5へ表示される。モニタ装置5には、このビデオ信号に基づいて画像が表示される。

サウンドブロック12は、PCM方式あるいはFM方式による音声合成を行うDSP140と、DSP140を制御等するCPU141により構成されている。DSP140によって生成された音声データは、D/Aコンバータ170により2チャンネルの信号に変換された後に2つのスピーカ5a又は5bに出力される。

サブシステム13は、CD-ROMドライブ等を搭載し、CD-ROM等の記録媒体によって供給されるアプリケーションソフトの読み込み、動画の再生等を行う機能を備える。

10 以上説明した画像生成装置によって行われる処理について説明する。図2は、CPUブロック10によって実行されるメイン処理を示すフローチャートであり、キャラクターを円運動させるための円運動処理S200、仮想的摩擦処理S202高低差がある地面への影の表示(S204)：すわなちキャラクターの投影表示処理、交差判定処理(S206)、キャラクターのモーション処理制御(S208)、キャラクターの高低差処理(S210)が順次行われる。必要が無い処理は、行われることなく次の処理に移行する。  
15 各処理の内容を以下に詳説する。

#### 円運動処理：

この処理は、概ね次のような内容である。既述のコントロールパッド2bの十字状に形成された方向キー26b(図26をも参照)を図示左又は右に操作すると、この方向キーによって操作されるキャラクター(闘士)30(図3参照)が仮想空間内を上下左右方向に移動する。円運動処理モードでは、遊戯者が操作する闘士30と相手側の闘士32との間に仮想的に向心力(相手側のキャラクターに向かう力)が与えられ、遊戯者側の闘士30が相手側の闘士の回りを自動的に周回(円運動)する処理30が実現される。

25 この処理内容を、図4のフローチャートに基づいて説明する。ステップ400において、円運動処理モードに入った否かが判定される(判定1)。遊戯者(プレーヤ)が所定の操作を成した場合に円運動要求があったとされる。例えば、コントローラ(パッド

2 b) の所定の釦を押下した場合である。次に、図 2 6 の 2 6 b の方向キーの図示左右方向操作量が読み込まれる等して、遊戯者の闘士と相手側の闘士との位置とが検出され、相手側の闘士が遊戯者側の闘士に対して成す方向が計算される（ステップ 4 0 2）。

次いで、遊戯者側の闘士 3 0 にその方向の角度が与えられて、かつ、この闘士を相手側の闘士に向かせる仮想的な向心力（求心力）が与えられる（ステップ 4 0 4）。この求心力を、物理的に考えれば、求心力は 2 つの物体間に働く引力に等しい。

相手側の闘士の x 座標と z 座標とをそれぞれ (e x p, e z p) とし、相手側の闘士の x 座標と z 座標とをそれぞれ (m x p, m z p) とすると、既述のような相手側闘士 3 2 の方向は、 $\arctan(e x p - m x p, e z p - m z p)$  によって計算される。なお、この計算に際して、両方の闘士の高さ方向の y 方向の座標は、便宜上考慮していない。

この結果、求心力と横方向キーによる左右方向の速度によって、仮想空間内の円運動が作られる（ステップ 4 0 6）。この円運動によって遊戯者の闘士 3 0 が相手側の闘士 3 2 を向きながらその回りを回るように動く。図 5 は、この円運動を仮想空間の上方より見た概念図であり、闘士 3 0 が状態 1 から状態 2、状態 3 を経て状態 4 までの円運動を行う。既述の図 3 は、状態 1 に係わる仮想空間を所定の視点から表した映像であり、既述のモニタ 5（図 1 参照）に表示される映像である。図 6 は、状態 4 に係わる、図 3 と同様な映像である。

したがって、説明された円運動処理によれば、一つのキャラクタに仮想的な向心力を与えているために、遊戯者にとっては、図 2 6 の方向キー 2 6 b を図示左右方向に操作するだけで、容易にキャラクタ 3 0 に円運動を与えることができる。例えば、既述のように相手側闘士 3 2 の回りを自己が操る闘士 3 0 が周回する動きを簡単に実現できる。

#### 仮想的摩擦処理：

この処理は、キャラクタの動きに、地面の傾きによる滑りと、地面の性質によってキャラクタに及ぼす摩擦力を変え、かつ、キャラクタが移動しているときには動摩擦を与え、キャラクタが移動していないときには静摩擦を与えることにより、キャラクタの動

きをより多彩にすることが内容である。

先ず、図7は、仮想空間内の地面の断面を示したものである。図8に示すように、ステップ800において、地面の傾きから滑り量を次のようにして計算する。

$$\text{滑り量} (= dv / dt) = i v g$$

5  $i$  : 所定の定数

$v$  : 地面の傾きを表す法線単位ベクトル

$g$  : 重力加速度

ここで、法線単位ベクトルとは、図7(1)に示すような法線ベクトルの地面の接線方向へのベクトルである。

10 この法線単位ベクトルが仮想空間の座標軸(x軸)に対して成す角( $\theta$ )が大きくなると、この地面上に在るキャラクタに対して、傾いた地面に沿って下がりようとする滑り量(加速度)がより大きく与えられる。

次のステップ802においては、キャラクタ(闘士)が移動しているか否かが判定される。例えば、遊戯者側から云うと、闘士の動きを制御するための方向キーが操作されているか否かが判定される(判定1)。

キャラクタが動いている場合は、動摩擦が与えられ、キャラクタが動いていないときは、静止摩擦が与えられる(ステップ804、806)。摩擦( $= dv / dt$ )自体は、次のようにして計算される。

$$M \cdot (dv / dt) = \mu \cdot S \cdot M (g \cdot \cos \theta + (dv' / dt) \cdot \sin \theta)$$

20  $dv / dt = T \cdot g \cdot \cos \theta + T \cdot (dv' / dt) \cdot \sin \theta$

$M$  : 重さ

$v$  : 摩擦速度(計算後の物体の速度)

$v'$  : 物体の速度

$\mu$  : 摩擦係数

25  $S$  : キャラクタの接地面積(定数)

$g$  : 重力加速度(定数)

$\theta$  : 地面の角度

$$T = \mu \cdot S$$

動摩擦及び静止摩擦は、この式によって得られた摩擦（或いは摩擦係数）に所定の処理を与えることにより決定される。但し、静止摩擦は動摩擦より大きな値となる。この摩擦は地面の角度によって変化するものであり、地面の傾きが大きいほど摩擦は小さい値となる。

次に、地面の属性が決定される。すなわち、キャラクタが接地している地面が、例えば、水場或いは砂場であるかが判定される。これらの属性は、予め地面毎に仮想的に与えられている。例えば、砂場では計算された摩擦量を2倍とし、水場では摩擦量を3倍する。次いで、先に計算された滑り量に、この摩擦力を減じるように反映させる（ステップ810）。

この処理によって、静止摩擦あるいは動摩擦がキャラクタに常に与えられるために、止まっているキャラクタはなかなか動き出せず、一度動き始めるとより移動し易くなる。また、地面の属性によってキャラクタの移動特性を変えることができる。これらのことにより、キャラクタの移動をより現実感豊かに表現することが可能となる。

高低差がある地面への影の表示：

図9にあるように、この処理はキャラクタ91が立っている地面90に凹凸92がある場合、キャラクタの影（投影像）をこの地面に凹凸をつけながらより簡単な計算によって表示することを内容とする。図9はこの処理の概要を示す。図中の94は高さ0の仮想的な基準線であり、そして、96は無遠慮にある平行光源である。

この人物は、頭、胸、腹、腰、足等のオブジェクトの集合構成されている。各オブジェクトは、複数のポリゴンによって作られている。図面のyGは基準線からキャラクタが立っている地面迄の距離であり、y1・・・ynは、各オブジェクト毎の地面から基準線までの距離である。

この処理は、次の計算式によって演算される。

$$E'(x', y', z') = Ms \cdot E(x, y, z)$$

$E'(x', y', z')$  : 影の座標点



$E(x, y, z)$  : 人物（キャラクタ）の座標点

$M_s$  : 影の座標をワールド座標系に変換するためのマトリックス（行列式）である。

この $M_s$ は次のようにして与えられる。

$$M_s = M_u \cdot (-T_{y_g}) \cdot P_n \cdot T_{y_n} \cdot (-T_{s_n}) \cdot R_n \cdot T_{s_n}$$

5  $M_u$  : 人物の座標を図9のワールド座標系に変換するための変換マトリックス

$T_{y_g}$  : 人物が立っている地面から基準線まで平行移動させるための平行移動マトリックス

$P_n$  : 人物を基準線に斜投影するための変換マトリックス

10  $T_{y_n}$  : 斜め投影された影を各オブジェクトの接地点（即ち、各オブジェクトが置かれる地面）迄平行移動させるためのマトリックス

$T_{s_n}$  : 各オブジェクトをワールド座標系の原点に対して平行移動させるための平行移動マトリックス

$R_n$  : 地面の傾きに合わせて影のポリゴンを回転させるためのマトリックス

15 この処理は、次の過程によって行われる。

処理過程1 : 影を作るための人物類似のキャラクタ91が地面から高さ0の基準線まで平行移動され、次いで、

処理過程2 : この人物類似のポリゴン集合体91を、前記平行光源96に基づいて基準線94に斜投影し、次いで、

20 処理過程3 : 斜め投影されたオブジェクトを各オブジェクト毎に  $y_1 \cdots y_n$  の幅で地面までそれぞれ平行移動させ、次いで、

処理過程4 : 各オブジェクトをワールド座標系の位置からその系の原点まで平行移動させ、次いで、

処理過程5 : それぞれの地面の角度に応じて、各オブジェクトを回転させ、最後に、

25 処理過程6 : 処理過程4によって原点まで移動したオブジェクトを元の座標系に戻すための平行移動が行われる。

したがって、この処理によれば、地面90に凹凸がある場合、この凹凸に合わせてキ

キャラクタの投影像を凹凸を付けて簡単に地面 90 上に表示することができる。

交差判定処理：

図 10 は、二つのキャラクタ、例えば、互いに向き合う闘士（人物 1，人物 2）を把  
 5 える仮想カメラの視野領域を示す x z 平面図である。この仮想カメラ（視点）によって  
 既述の図 3 に示すような映像が表示される。この処理においては、この視野領域に闘士  
 以外の壁、建屋等視野領域を遮るような仮想の構造物が重なった場合、この構造物の一  
 部或いは全体を表示しないようにされる。図 10 において、構造物 100，102 は、  
 視点から人物に至る視野領域を遮るのに対して、構造物 104 は視点によって表示され  
 10 る領域にあるものの前者の領域を妨げない。ここでは、構造物 102 と 104 とを消去  
 する画像生成或いは画像処理が行われる。

図 11 は、この処理を示すフローチャートである。この処理に際して、これらの構造  
 物が視野領域に重なっているか否かの判定をより迅速かつ容易に行うために、x z 座標  
 面において判定が行われ、かつ厚さを持ったものは、円によって近似される。図 12 は、  
 15 この構造物を内接円で近似した状態を示した x z 平面図である。

図 11 において、ステップ 110 は、この近似円が前記視野の領域に重なっているか  
 否かを判定する処理である。図 13 はこのステップにおける判定の概念図である。視点  
 から人物 1，2 に至るまでの視野領域に相当する三角形の領域を構成する全ての辺につ  
 いて、中心点 T の半径 d を持つある近似円が重なるか否かの判定を以下の処理に基づい  
 20 て行なう。なお、以下の説明において、ベクトル L、ベクトル R、ベクトル T は、所定  
 の原点から L、R、T に対して成すベクトルである。なお、三角形の各辺のそれぞれの  
 頂点について時計方向に L 点、R 点が順に設定される。

【数式 1】

$$\begin{aligned}
 \vec{r} &= \vec{R} - \vec{L} \\
 \vec{t} &= \vec{T} - \vec{L} \\
 \vec{p} &= \vec{c} + \vec{L} - \vec{T} \\
 cx &= rx \cdot \frac{|\vec{c}|}{|\vec{r}|}, \quad cy = ry \cdot \frac{|\vec{c}|}{|\vec{r}|}
 \end{aligned}$$

ここで、

5 【数式 2】

$$|\vec{c}| = |\vec{r}| \cos\theta$$

$c_x$  : ベクトル  $c$  の  $x$  成分

10  $c_y$  : ベクトル  $c$  の  $y$  成分

$r_x$  : ベクトル  $r$  の  $x$  成分

$r_y$  : ベクトル  $r$  の  $y$  成分

内積の定理により、

15 【数式 3】

$$|\vec{r}| = |\vec{r}| \cos\theta = r_x t_x + r_z t_z$$

$$\frac{|\vec{c}|}{|\vec{r}|} = \frac{r_x t_x + r_z t_z}{|\vec{r}|^2} = \frac{r_x t_x + r_z t_z}{r_x^2 + r_z^2}$$

20 但し、

【数式 4】

$$0 \leq \frac{|\vec{c}|}{|\vec{r}|} \leq 1$$

25 近似円の中心  $T$  が直線  $TR$  の正領域にある場合（三角形の内側より外に  $T$  がある場合）、

【数式 5】

$$|\vec{p}| - d > 0$$

差 替 え 用 紙 (規則26)

ならば、円は三角形の内側の領域に対して独立、すなわちこの領域に重なることはなく、

5 【数式 6】

$$\frac{|\vec{c}|}{|\vec{r}|} < 0 \quad \text{または} \quad 1 < \frac{|\vec{c}|}{|\vec{r}|}$$

の時は、点 T が直線 L R の正領域にあれば、円は領域に対して独立となる。

- 10 ステップ 1 1 0 において、近似円が三角形の領域内に重なっている場合は、この構造物が表示されないように処理され（ステップ 1 1 1）、重なっていない場合は、次の処理（ステップ 1 1 2）に進む。次の処理は、この構造物が一定方向に長さを持った壁の如くのものについての重なり判定を行うことを内容とする。この処理は、図 1 4 に示すように次の如く行われる。

- 15 直線 T S として近似された構造物について、この直線と三角形の領域との交差（重なり的一种）を判定する。三角形の領域を構成する全ての辺について、以下の処理を行う。直線 L R と任意の点 P との位置関係を判定する関数を F 1 (P) とすると、

【数式 7】

$$\begin{aligned} F_1(P) &= a_1 P_x + b_1 P_z + c_1 \\ a_1 &= L_z - R_z \\ b_1 &= R_x - L_x \\ c_1 &= L_x R_z - R_x L_z \end{aligned}$$

直線 S T と任意の点 P との位置関係を判定する関数 F 2 (P) とすると、

25 【数式 8】

$$\begin{aligned} F_2(P) &= a_2 P_x + b_2 P_z + c_2 \\ a_2 &= S_z - T_z \\ b_2 &= T_x - S_x \\ c_2 &= S_x T_z - T_x S_z \end{aligned}$$

5  $F1(S) \cdot F1(S) = 0$ かつ、 $F2(S) \cdot F2(S) = 0$ ならば、直線TSは直線LRに交差していると判定される。

この交差が否定された時は、ステップ114に移行し近似円の重なりの有無が判定される。この理由は次のとおりである。三角形の領域内に既述の直線TSが完全に入ってしまった場合、重なり判定が否定されて壁状の構造物が表示されてしまう。そこで、この直線TSに図15のような壁の厚さに相当する小円150を与え、この小円について  
10 ステップ110と同じ様な重なり判定を行う。この結果、直線TSが完全に三角形の領域内に入ってステップ112の判定において重なり無しと判定された場合でも、この近似円が三角形の領域内に入っている場合、重なりあり、と判定されてステップ111に移行する。

これらの処理により、仮想カメラから表示対象となっているキャラクタに向けた視野  
15 領域に壁、柵、塀等の構造物が表示されないようできる。そして、図10に示すように、この壁等の構造物がこの視野領域から出ている104のような場合は、そのままこの構造物が表示される。以上の処理によれば、余計な構造物によってキャラクタを遮ることなく好適にキャラクタを表現することができる。もっとも、表示対象としてのキャラクタが設定されていない場合は、これらの構造物はそのまま表示される。なお、近似円と  
20 しては、図12の場合の内接円に代えて外接円でも良い。

#### キャラクタのモーション処理制御：

ここでの処理は、一つのモーションから他のモーションをつないで一連に表示させようとする場合で一つのモーションの終了位置を目標点に合わせるように補間すること  
25 を内容とする。

図16は一つのキャラクタのモーションの一連の過程を示すものであり、人物類似のキャラクタ（闘士）160がプレイヤーによるパッド2bの操作によって相手側の闘士に

向けて、腕部 160A による殴打を繰り出す過程が示されている。

この腕部の挙動は、公知のスプライン関数（特に、3 次スプライン関数）によって計算されモニタに表示される。今ここで、この腕部 160A が相手側闘士の胸部に向けて移動しているとして、ある時点で相手側闘士が y 方向に移動した場合（例えば急に屈んだ場合）、遊戯者にとってみれば、移動している腕の軌跡を急激に図 17 の如く下方に変えることは非常な困難を伴うし、また、このことを CPU ブロック 10 の側で単に行おうとすると腕の動きが不自然なものになり易い。

そこで、現在実行中のモーションのスプライン軌跡変化過程の終了付近につき領域を設け、このつながり領域に、当初のスプライン軌跡の終了点が別な位置の目標点になるように、新たなスプライン関数を与えるなどの補間を行うようにした。

このことを図面を用いて説明する。図 18 の（1）に示すように、図 16 の闘士の腕部の先端にある手の甲 180A は、スプライン関数によって定められた軌跡 180 に沿って移動する。図 18 の（2）は、このスプライン関数によって得られた軌跡 180 を示したものであり、手の甲 180A はこの軌跡の開始点から移動して終了点に至り、その後開始点まで復帰する。

今、手の甲 180A がスプライン関数軌跡 180 の開始点からその終了点に向けて移動しているとする。この過程で、例えば相手側闘士の挙動変化等によって当初の終了点を目標点に変更する必要があるとする。

そこで、CPU ブロック 10 は、手の甲の位置 180A がつながり領域にある時点で目標点において終了するように、この領域内の関数を補正する。この補間は、例えば、後述のように行われる。勿論、つながり領域のスプライン関数の次数や係数等を適宜変えることによって可能である。その結果、図 16 に示すモーションが図 17 に示すモーションに自動的に補間されるし、つながり領域におけるモーションもより円滑に表現される。なお、図示する上での便宜上、図 18 の（2）において縦軸は y 軸方向の移動量のみを示している。3 次元ビデオゲームの一般論のとおり、x 軸とはモニタの画面に臨んで横方向に与えられ、y 軸はその高さ方向に与えられ、z 軸はその奥行き方向、すなわち画面と直交する方向に与えられる。

つなぎ領域（つなぎ時間）はエンドフレームを備えた複数のフレーム群によって適宜設定される。また、手の甲以外の上腕部及び下腕部の各オブジェクトについても、同様な処理が適用されて腕全体の補間されたモーションが完成される。つなぎ領域における補間された座標は、次の特性式によって設定される。

5 
$$\text{現在座標} + (\text{現在座標} - \text{目標座標}) \cdot (\text{現在時間} / \text{つなぎ時間})$$

図19は以上の処理を説明するためのフローチャートであり、ステップ190において、遊戯者がパッド26bを操作してモーション移動のコマンドが発生したか否かが判定される。ステップ192では、終了点と異なる座標に目標点が設定されたか否かが判定される。ステップ194では、選択されたモーションが再現される。ステップ196  
10 では、この判定が肯定された場合、つなぎ時間が計算されるとともに、ステップ194に移行しこのつなぎ時間の間補間されたモーションが再現される。ステップ198では、現在のフレームが一連のモーションのエンドフレームか否かが判定され、エンドフレームである場合には、メインルーチンにリターンし、エンドフレームに未だ至っていない場合には、ステップ192に戻る。

15 ここで、つなぎ時間を成すつなぎフレームはモーションの最大フレーム数（図18（2）のスタートフレームからエンドフレームまでのフレーム数）の0.1であり、つなぎフレームが10を越える場合は10に固定される。つなぎフレームは、1以上、最大フレーム数の1/2の範囲であれば良い。つなぎフレームが余り長いとモーションが持っている元々の動きが失われ、一方、余り短いと補間が急峻となり動きに円滑さが失  
20 われる。

この処理によれば、次のような効果が達成される。ビデオゲーム装置のように、ゲーム画面に登場する闘士等のキャラクタを素早く操作するためにコントローラが操作されると、キャラクタの多彩な動きが連続的に再現される。このとき、一連のモーションが再現されている時点で、相手のキャラクタの急な動作に合わせてそのモーションを変  
25 化させようとしても、モーション自体が固定されたものである限りそれは困難である。しかしながら、既述のようにモーションが補間されると、例えば相手側キャラクタの動きに合わせたモーションを多彩に再現できることになり、モーションがより現実感豊か

なものになる。通常、このような補間をユーザが行うことは極めて困難であり、なによりも、従来のビデオゲーム装置ではこのような補間を行うようには構成されていない。

なお、この処理は、スプライン関数によってモーションが計算される場合を例にとって説明したが、予め決められたパターンが順次再現されるパターンチェンジの方式に、  
5 この処理を適用することも可能である。この場合は、つなぎ領域において補正されたパターンを再現するようにすれば良い。

キャラクタの高低差処理：

この処理はキャラクタ（闘士）に互いに地面からの高低差があるときに、この高低差  
10 を補正して所望の画像を表示することを内容とする。例えば、図20に示すように、互いに向き合う二人の闘士200、202との間に高低差が存在する場合、闘士200からパンチを高い位置にある闘士に向けて水平に繰り出すことは現実的ではない。そこで、この処理は、この高低差を補正して、闘士200からの殴打を高い方向に向けて繰り出すようにさせた（図23参照）。

15 この処理は、図21に示すように、ステップ2100において、この高低差を補正する必要があるか否かが判定される。この判定において「必要あり」とされるのは、例えば、図のように互いに向き合う闘士について、一方の闘士から他方の闘士に対して、腕や足による殴打の攻撃要求（コマンド）が発生した場合である。

次に、この判定が否定された場合は、メインルーチンに復帰する。一方、この判定が  
20 肯定された場合は、さらに、ステップ2102において、相手側闘士が地面上に立っているか、すなわち、キャラクタと地面との衝突判定が行われ、これが肯定された場合には、キャラクタの高低差処理を必要として次にステップに移行する。この判定が否定された場合は、図のルーチンを終了する。

次のステップ2104では、遊戯者自身が操るキャラクタが立っている地面からのこの  
25 のキャラクタまでの距離を演算する。相手側のキャラクタについても同じ演算を行う。次に、ステップ2106において、この結果の値について所定値と比較し、所定値の範囲を越える場合は、ステップ2108でこの範囲内に前記距離に関する差分を収める処



理を行う。すなわち、一方の遊戯者が操る側の闘士の高さ（ $m_y$ ）と他の遊戯者が操る側或いは画像処理装置自体が所定のプログラムによって自動的に操る側の闘士の高さ（ $e_y$ ）とすると、両者の差分（ $d i f f 1$ ）は、（ $m_y - e_y$ ）となり、例えば、 $d i f f 1$ が、

5  $-0.15 \leq d i f f 1 \leq 0.15$  の範囲内か否かが判定され、 $d i f f 1$  がこの範囲を越える場合は、この範囲より小さい  $d i f f 1$  は、これを  $-0.15$  とし、この範囲より大きいものは、これを  $0.15$  とし、この範囲内である場合は、 $d i f f 1$  をそのまま差分結果（ $d i f f 2$ ）として決定する。このように  $d i f f 1$  に対して補正を行う理由については後述する。

10 次のステップ 2110 では、 $d i f f 2$  を攻撃力が発生する時間（フレーム数）で割り、この結果を「結果 1」とし、 $d i f f 2$  を攻撃力発生から硬化が解けるまでの時間で割ってこの結果を「結果 2」とする。なお、このことを図 22 を用いて説明する。図 22 は、攻撃技（闘士が繰り出す蹴り、パンチ、ジャンプ等）の流れを示したものであり、1、2・・・5 の各時間毎のフレームの流れ（時間の流れ）に対して次のようである。

15

- 1：技の開始（0 フレーム目）
- 2：攻撃力の発生（1 から 3 フレーム目迄）
- 3：攻撃力の消滅（3 から 4 フレーム目迄）
- 20 4：硬化時間（これは他の攻撃技のコマンドを受け入れない時間、0 から 4 フレーム迄）
- 5：技の終わり（0 から 5 フレーム目迄）

したがって、既述の「結果 1」は、（ $d i f f 2$ ）／2 となり、  
「結果 2」は、（ $d i f f 2$ ）／（4－2）となる。これらの値は、当然のことながら、  
25 所定のワーク RAM 領域に設定記憶される。図 22 に示す特性は各技毎に予め決められている。結果 1 は攻撃力発生迄のフレーム毎の変化率に相当する値であり、結果 2 は、硬化が解ける迄のフレーム毎の変化率に相当する値である。

次いで、ステップ2112では、要求された技毎に技を繰り出す際の手や脚等が移動する座標を決定する。この座標値は、相手側のキャラクタに対して高さの差が無いことを前提に予め決められている。この座標（リーフ）を「結果3」とする。このリーフは、リーフを返すための関数 *fat* によって技を繰り出す際の手や脚の空間座標位置に応じて決定される。次のステップ以降は、攻撃技を実行する際の処理を説明するものである。

ステップ2114において、技が発生してから現在表示されるフレームが何フレーム目に相当するかが検出される。次のステップ2116では、このフレームが攻撃力が発生する以前のフレームであるか否かが判定される。図において、フレーム1迄は攻撃力が発生する以前のフレームであると判定される。

次にステップ2116における判定が肯定された場合、ステップ2118において、前記「結果1」とフレーム数を乗じ、この演算結果を差分として前記ステップ2112のリーフに加算する。そして、この結果を用いて公知のインバースキネマテックの技術（例えば、「ロボットの力学と制御」、システム制御情報学会編、有本卓著、朝倉書店発行、5. 2 逆運動学（132頁～））を用いて、リースにこの差分が加えられた値等に基づいて闘士の体の形を再計算する。

既述のステップ2116において、攻撃力が発生した以降であると判定された場合は、ステップ2200において、硬化が解けるフレーム数（図の「4」）から現在のフレーム数（図の「2又は3」）を減じ、結果2にこのフレーム数を乗じてステップに移行する。

この結果、ステップ2118によって図23に示すように自己が操る闘士200から相手の闘士202に向けて、闘士の高低差を補償した高さにパンチを繰り出し、パンチの命中後或いはパンチが命中しないとしても、その後パンチとして繰り出された腕を高低差を補償しながら元の位置まで帰還させる。

ここで説明した処理によれば、二つの闘士の高低差を反映させた画像処理が実行されるので、一つの闘士から繰り出した攻撃（パンチ、キック）が高い位置に在る闘士に向けて行われ、実際の攻撃の現状に即した映像を提供することが可能となる。したがって、

高い位置にある闘士に対するモーションを事前にメモリに記憶していなくても、一つのキャラクタと他のキャラクタとの間の画像処理を、キャラクタ間の高低差を反映した状態で実現することができる。

5       なお、diff1に対して補正を行ったのは、次の理由による。このような補正を行わないと、両方の闘士の間に大きな高低差があると、一方の闘士から他方の闘士に向けた殴打が、極端に云うと地面に対して直角方向に向けられてしまい、かえって不自然になるからである。

10       なお、既述の各処理の説明において、ビデオゲームを例にとり説明したがこれに限定されるものではない。図1において、CD-ROMに代えてROMカセットを用いても良い。これらのものは本発明に係わる画像生成装置の動作プログラムが記憶された記憶媒体として機能するものである。また、交差判定処理において、既述の構造物を完全に表示しない場合の他、この構造物をメッシュポリゴン或いはポリライン等で表示することも、「表示しない」の態様に含まれることとする。なお、図24及び図25は、本発明のモデルとしての人物類似のキャラクタ（闘士）のそれぞれ一例である。

15       CPUブロック10によるグローシェーディング（その他のシェーディングについても同様）に当たっては、キャラクタの全体ポリゴンについて行う場合のほか、キャラクタの一部、特に色を線形補間して立体的に見せたい部分（例えば、露出している皮膚に相当するポリゴン）のみについて処理を行うすることもできる。こうすることにより、CPUブロック10に与える処理遂行上の負荷を低減することが可能となる。図24の  
20       キャラクタでは衣服より露出している手の甲、顔、そして首下の胸部についてのみグローシェーディングが行われ、図25のキャラクタでは、ほぼ上半身並びに脚部についてのみグローシェーディングを行うようにしてもいい。

25       また、各ポリゴンに対するZソートをかける場合、4角形のポリゴンでは従来より4頂点を利用してはいたが、これを2頂点をそれぞれ結ぶ対角線の交点（中点）のみで判定するようにしても良い。この結果、メインCPUがメモリをアクセスする頻度を減少させて処理速度を向上させることができる。なお、前記中点によってZソートをかけてもこの中点は4頂点の平均値であるため、4頂点を用いたZソートの精度をそのまま維持

することができる。三角形のポリゴンの場合は、重心を利用すれば良い。

次に、既述の画像生成装置の処理動作の他の実施例について説明する。図26は、前記コントロールパッド2bの詳細を示す正面図であり、このものは、既述の方向キー26bとA、B、C、X、Y、Z、L、Rの各ボタンを有している。これらの方向キーや  
5 ボタンの押下は、図24或いは25のキャラクタ（闘士）の動きに対応しており、例えばAボタンは「相手闘士からの攻撃に対する防御行動」、Bボタンは「相手側闘士に繰り出すパンチ」、Cボタンは「相手側闘士に向けて繰り出すキック」に相当する。

ユーザである遊戯者はこれらのキーやボタンを多彩に操作して自己が望むようにキャラクターを操作しようとするが、極めて迅速かつ多彩に動く闘士を適切に操作するには  
10 は非常な熟練を要する。

そこで、ここで説明する処理の形態においては、図27に示すように、複数のボタンを押下することの組み合わせであるコマンド技の集合を混合技とし、これをファイルとして予めワークRAMに設定する。そして、これを既述のボタンのうちの一つを押下することによって読出し、闘士がコマンド技を順に繰り出すようにしている。図27において、PPKのようにあるのは、Bボタン、Bボタン、Cボタンが順に押下された場合に等しく、P+Kのようにあるのは、BボタンとCボタンが同時に押下された場合に等しい。さらに、各コマンド技についてその技が占めるフレーム数が設定できるようになっている。したがって、遊戯者は混合技中の各コマンド技のフレーム数を自由に設定できるために、自己が望む混合技を多彩に作り出すことができる。この混合技の具体例を  
15 述べれば、「相手側の闘士を巴投げし、起き上がったところに蹴りを与える」等である。

さらに詳細に説明すれば、「P、K、P」は、パンチボタン、キックボタン、パンチボタンが連続して押されることを示し、且つ、これらが、連続されたと判断されたときは巴投げをすると仮定すれば、Pが押されたときに、キャラクタはパンチのモーションに入る。このモーションの始まりから、例えば、モーションがパンチ戻りになるもまでの間に、Kが押されたとすれば、繰り出されたパンチの姿勢から、パンチ戻りのモーションになるタイミングで、キックが始まり、キックのモーションが繰り出される間にさらに、Pが入力されたときは、連続技と判断して、巴投げのモーション（コマンド技）  
25

に連続していく。この時のタイミングを、先のキックで、相手のキャラクタが倒れかかっているタイミングに合うようにすれば、同じ巴投げでも、技が掛かり易い等、非常に効果的な巴投げを作ることができる。

5 遊戯者は、これらの混合技ファイルを複数作り、各ファイルに一つのボタンを割り付けることができる。もっとも、この混合技ファイルが割り付けられていないボタンを押下すれば元もとのボタンに割り付けられている単独技が繰り出される。どの範囲のボタンに混合技を登録するかは、混合技の必要性或いは単独技の必要性に応じて適宜決定される。各ファイルに割り付けるボタンは、複数であっても良い。

10 次のこの処理の動作を図28に示すフローチャートに基づいて説明する。先ずステップS280において、既述のボタンが押下されたか否かが判定される。ボタンが押下された場合、押下されたボタンに対して混合技が登録されている場合（ステップS282肯定）、ステップ284において、ワークRAMに登録されている混合技中の各コマンド（複数或いは単独のボタンの押下に等しい）が順に読み出される。次いで、ステップS286において、ファイル中のコマンドが最後まで実行されたか否かが判定され、こ  
15 れが肯定された場合はこのフローチャートが終了されるまで繰り返し実行される。

一方、ステップS280においてボタンの押下が無い場合はリターンし、また、ステップS282において押下したボタンが混合技のファイルに登録されていない時には、そのボタンに割り付けられた単独技が実行される。

20 ここで説明した形態によれば、キャラクタ（モデル）に対する動作コマンドの組み合わせ（例：動作スイッチを押下する組み合わせ：一つのスイッチの押下、複数のスイッチの押下、これらスイッチを順に押す、あるいはこれらスイッチを同時に押す）を予めメモリに記憶させ、これを簡単なスイッチ操作、例えば、一つのボタンを押下すること、によって、画像処理手段が読み出し、読み出されたコマンド群に基づいてモデルの動き（モーション）を連続的に制御するようにしている。したがって、既述のように、  
25 ユーザが複雑なキー操作を行うことなく、かつ自己の好みに応じたより多彩なモデルの動き（モーション）を表現することができる。

なお、図29は、ここで説明した混合処理の際に、モニタに表示されるモードセレクト

ト（モード選択）のための初期画面であり、混合技の選択のためには上段右端のアイコンが選択される。図30は、ボタン（キー）を技に割り当てた

（Assign）したことを示す画面、図31は、前記混合技ファイルを作成するための画面、図32は、互いの闘士が格闘状態にあることの1状態を示す画面である。図33はこの  
5 1状態の画面にさらに後に展開される画面であり、闘士（「あきら」）が闘士（「ラウ」）に倒された状態のものである。なお、図33から明かなように、一方の闘士が他方の闘士を倒したとき（CPUブロック側でそう判定されたとき）、画面中のキャラクタ330がそれを表示するようになっている。図33においては、「つぼみの植物類似のキャラクター」が「花がさいた植物類似のキャラクター」に変更されることによってこのこと  
10 を表示するようになっている。

## 第2の実施例

続いて、画像処理装置としてのビデオゲーム機の第2の実施例を図34～図68に基づき説明する。なお、この第2の実施例におけるビデオゲーム機のハードスエア構成は前述した第1の実施例のものと同一であるので、その説明は省略する。

15 図34は、CPUブロック10によって実行されるメイン処理を示すフローチャートであり、後述の各処理が実行される。S200は残像処理ルーチンであり、S202は舞上げ処理のルーチン（飛翔体の処理）であり、S204はモデルの自由落下運動処理、S206は不定形区画とモデルとの衝突判定処理、S208は再生速度の変更処理を行う。これらの各処理は、繰り返し実行される。以下、各処理について説明する。

20

## 残像処理

図35は、モデルとしての闘士に対して適用される残像処理の原理を示す。図35において、30は闘士であり、闘士の脚30Bについて残像が表示されている。すなわち、現在のフレーム（コマ）の実像が表示されると同時に、以前のフレームの残像1乃至4  
25 が同時に表示されている。

図36はこの残像処理の原理を示すものである。ポリゴンデータファイル（或いはパラメータファイル）は、この闘士の脚の「廻し蹴り」モーションを規定するパラメータ

を記憶している。ポリゴンデータファイルの# 1乃至# nは、残像開始時点からのコマ数を示している。

ポリゴンデータファイル# 1には、残像4に相当するモデル# 1が記憶されている。従って、この時点では、画面上に闘士のモデル# 1が実像として表示される。次の時点では、ポリゴンデータファイル# 2に、残像3に相当するモデル# 2が記憶される。したがって、この時点では、画面上に闘士のモデル# 2が実像として表示される。以後同じようにして、画面上に闘士の脚モデル# 3・・・、# nが表示される。図4の4コマ目の場合（# 4）は、脚の実像モデルとしては、# 1乃至# 4までが順次表示される。

残像用バッファとして、B # 1、B # 2、B # 3、及びB # 4が設定されている。これらのバッファに、ポリゴンデータファイルに記録された闘士のモーションデータが順次記録される。図36の下欄にはこのことが示されている。ポリゴンデータファイル# 1に記録されているモデル# 1のデータは、残像開始後残像用バッファ# 1に記録され、次の時点では、ポリゴンデータファイル# 2に記録されたモデル# 2のデータは残像用バッファ# 1に記録され、残像用バッファ# 1の内容が残像用バッファ# 2に送られる。すなわち、残像用バッファのデータは、# 1→# 2→# 3→# 4の順序に送られて、残像用バッファ# 1にはポリゴンデータファイルから一つ前のコマにおける実像のモーションデータが送られる。残像用バッファ# 4は、順次そのまま更新される。

例えば、5フレーム目の時点では、ポリゴンデータファイルにはモデル# 5のモーションデータが格納され、かつこのモデルが実像として表示されている。一方、残像用バッファ# 1にはモデル# 4のモーションデータが記憶され、残像用バッファ# 2にはモデル# 3のモーションデータが記憶され、残像用バッファ# 3にはモデル# 2のモーションデータが記憶され、残像用バッファ# 4にはモデル# 1のモーションデータが記憶されている。

ポリゴンデータファイルのモーションデータと、残像用バッファのデータ（過去計算されたモーションデータに等しい。）とは同時に表示されるので、画面上には実像と残像とが同時に表示されることになる。図35はこのことを示している。実像に対しては通常のレンダリングが施されて表示されるのに対して、残像に対しては、演出効果を高

めるために、換言すれば「残像らしく見せる」ために、半透明の処理を施しながらレンダリングする。この半透明の程度をより前の時点のコマ数における残像ほど高めることができる。前のフレームの残像ほど透明に表示される。なお、残像用バッファは、RAM 102に設定される。

次に、この実施例における残像処理の動作について説明する。図37は、この処理を行う概略フローチャートである。まず、残像が必要であるか否かが判断される(S500)。この判断は、例えば、モーション毎に行われる。例えば、闘士の大きな動きが伴う大技(廻し蹴り、背負い投げ等)には残像が必要として処理される。図35を参照されたい。その他に、動作量が所定値以上の場合も同様である。

図38は、この残像必要性処理の詳細フローチャートである。まず、技の発生がチェックされる(S600)。このチェックは、既述のコントローラ2bの操作ボタンや操作スティックの操作状況によって判定される。技が発生したと判定されると、技の属性データが読み込まれる(S602)。技の属性データとは、個々の技に対して与えられた性質に係わるデータであり、例えば、「攻撃の技か」、「脚による技か」、「手による技か」等である。

次に、この属性データが残像を発生すべきデータか或いはその逆かが判定される(S604)。例えば、「攻撃の技」、かつ「脚による技」の場合には、残像表現が必要であるとして、所定のフラグ“F”を立てて“1”とし(S606)、他の場合には、“F”を“0”とする(S608)。

概略フローチャートの次のステップでは、残像必要部位が特定される(S502)。この特定は、詳細フローチャートである図39に示す処理によって行われる。まず、S700において、攻撃コマンドが発生したか否かがチェックされる。攻撃コマンドとは、操作ボタンや操作スティックの操作が、遊戯者が操る闘士が対戦相手の闘士に対して、攻撃をするような命令態様に至った状態で発生するものである。

図39において、例えば、廻し蹴りのコマンドが発生した場合(図35の場合がこれに該当する。)、攻撃部位が足首のパーツ(図35の30A)であることが参照され(S702)、次いで、脚全体の残像を形成するために、足首30A、下肢30B、大腿部



30 A、すなわち、左側の脚全体（図35参照）が残像必要場所として特定される（S704）。なお、ここでの説明のように、闘士は複数のパーツから構成されており、各パーツはそれぞれポリゴンから構成されている。

もっとも、本実施例の場合のように、攻撃コマンドが発生していない場合にも残像を表示するようにすることができる（S706）。この場合には、当然のことながら、図37のS500における「残像が必要か？」の判定において、「現在攻撃中」でなくとも残像表現が可能となるように、残像の必要性について判定する。例えば、闘士が技を被って倒れる過程を残像を伴いながら表示するような場合である。この場合は、モーションデータから、S702、S704のように残像が必要なパーツを決定する。

そして、図37のS504では、図35に既述の残像表示を実行して、S506に至り、残像表示が終了したか否かが判定される。例えば、一連のモーションが終了したか否かが判定され、図35の例では回し蹴りが終了したか否かが判定され、終了したと判定された場合は、残像表示の処理が終了される。

この実施の形態によれば、闘士の一連の動きのデータを有するポリゴンファイルのデータをそのまま順次残像用バッファに記録している。このポリゴンファイルには、座標変換されクリッピングされたポリゴン各頂点の座標を各点での法線ベクトル付きで記録している。このポリゴンファイルのデータをそのまま残像バッファに記録して、これを現在の実像とともに出力するだけであるので、従来技術のように残像用のポリゴンをその都度座標変換（モデリング）する必要がなく、CPUブロックに与える計算負荷が軽減される。しかも、パーツ（脚）の過去の位置に残像が表示され、かつ、半透明の演出が加わっているために、残像表現を演出感豊かに表示することが可能となる。なお、半透明の演出に代えて、脚ポリゴンの表示をメッシュ状、あるいはラインによって表現するようにしても良い。半透明の処理によって、脚の残像と背景とが同時に表示される。

この実施形態によれば、残像が付されるような大技と残像が付されない中・小技とを残像の有無によって、遊戯者は視覚を介して区別できるために、ゲームの演出感（ゲームとしての画像表現の多様化、面白味等）が向上する。

## 飛翔体運動処理

次に、飛翔体（設置物）の舞い上げ処理について説明する。この処理は、後述のように、地面の砂、或いは水、また、地面に落ちている葉を舞上げる態様を備える。先ず、前二者のものについて説明する。その概要は次の通りである。

- 5 図35に示すように、闘士30が脚を用いて蹴り上げのモーションを採った場合、或いは、闘士が空中より着地した場合、水、砂を舞い上げる処理が行われる。ゲームの演出効果を高めるために、このような処理が実行される。特に、格闘用ゲームのような遊戯者が操作ボタンを激しく操作してキャラクタを迅速に操作しようとするビデオゲームにおいては、キャラクタの動きが多彩、かつ迅速であるため、このような水等の舞い  
10 上げをキャラクタの動きに合わせて迅速かつ的確に計算して表現する必要がある。以下に説明する処理がこのことを可能にする。

- 先ず、砂、或いは水の跳ね上げ処理について説明する。ここで、跳ね上げ処理とは、図40に示すように、パターンチェンジするポリゴン（全32パターン）が計算によって求められた軌道を順次運動することである。このパターンは、水或いは砂が順次拡散する状態をシミュレートしている。砂或いは水の一粒一粒を計算すると、計算時間が掛  
15 かるので数粒まとめてパターンチェンジしながら運動させている。

- この処理では、闘士のパーツチェックが行われる。前記闘士は、頭、右手（上腕、下腕、手首）、左手、胸、腹、腰、左脚（大腿部、下肢、足首）、右脚の各パーツから構成されている。パーツチェックは、脚、頭、腰に対して行われる。したがって、例えば、  
20 脚により蹴り上げ、頭或いは腰からの落下をシミュレートする。パーツチェックには、パーツの移動量のチェックが含まれる。

その他、仮想地面の属性フラグ、パーツ位置フラグがチェックされる。仮想地面属性フラグ(b water)が“1（水の属性）” or “0（砂の属性）”、及びパーツ位置フラグ(b air)が“1（着地状態）” or “0（空中状態）” に設定される。

- 25 (b water)が“1（水の属性）” の場合、水の跳ね上げが行われ、“0（砂の属性）” の場合は砂の跳ね上げが行われる。水又は砂の跳ね上げには、次の態様がある。

着地発生：パーツが地面或いは水面に着地（着水）した場合であり、発生点から砂或

いは水が四方に広がる。

蹴り上げ発生：パーツが地面（水面）に着地（着水）している状態で生じ、脚の蹴り上げによって砂、或いは水が脚（チェックしているパーツ）の移動方向に跳ね上げられる。なお、跳ね上げ発生、及び着地発生は、図60に模式的に示されている。図の280は、キャラクタ（闘士）であり、脚を蹴り上げて、砂或いは水282を脚の運動特性に応じて跳ね上げている。或いは、着地して水或いは砂を着地動作に応じて、四方に跳ね上げている。

この処理では、継続処理と称する、着地発生或いは蹴り上げ発生が数インターラプト（数コマ或いは数フレーム）の間継続される処理が実行される。通常、これらの発生時に跳ね上げ数を決定してその $1/4$ をセットする。跳ね上げ数を $N$ とすると、1インターラプト毎に次のように跳ね上げ数が順次減って行くようになっている。

跳ね上げ数を $N$ とすれば、

$$1 \text{ 回目 } 1/4 N, N' = N - (1/4) \cdot N$$

$$2 \text{ 回目 } 1/4 N', N'' = N' - (1/4) N'$$

$$15 \quad 3 \text{ 回目 } 1/4 N''$$

つまり、インターラプト毎に跳ね上げ数の残数の $1/4$ ずつセットして行く。跳ね上げの形態は、既述のようにパターンチェンジするようになっている。

着地発生の場合の処理について説明する。チェックパーツの前回（一つ前のインターラプト）の位置を $(OX, Oy, OZ)$ とし、今回の位置を $(PX, Py, PZ)$ とする。地面或いは水面の高さ（Y座標）を $epos$ とする。

$$\text{パーツの基本速度：} SPD_x = PX - OX \quad X \text{ 方向}$$

$$SPD_y = Oy - Py \quad Y \text{ 方向}$$

$$SPD_z = PZ - OZ \quad Z \text{ 方向}$$

跳ね上げ総数（セット数）： $amount = \text{基本セット数} \cdot \text{パーツ移動量}$

25 跳ね上げ原点（発生原点）：図41は、仮想空間の高さ方向（Y方向）より表した模式図である。チェックパーツが矢示の方向に移動しているとき、図42に示すように、水、砂の跳ね上げ発生原点は、 $(OX, epos, OZ)$ のように設定される。

砂（水滴）の発散処理：チェックパーツの基本速度のX、Z成分を抜き出す。図43は画像生成装置内に仮想的に設定されたモデル空間のX-Z平面である。なお、この図はこの平面を斜め上方の位置より見ている状態を示している。このX、Z成分のベクトルをXZ平面上で $+135^\circ$ 乃至 $-135^\circ$ の範囲でランダムに回転させる。図において、円で囲まれた範囲（ハッチング部分を除く）が砂（又は水滴）が拡散して行く方向である。

今、ランダムに回転させたベクトルをAとする。次に、このベクトルをベクトルの方向にL移動させてA'とする。ここで「L」は、パーツの半径である。例えば、脚の場合は10cm、腰の場合は20cm、頭の場合は15cmに設定されている。このA'に基本スピードのY成分を加算して最終的に砂、水滴の跳ね上げ速度が決定される。この速度が決定された後、図40に示す軌跡を採って砂、水滴等がパターンチェンジされながら移動される。よって、図44に示すように、パーツの回り（半径L）から砂（水滴）が四方に飛び散る状態がシミュレートされる。

次に跳ね上げ発生の処理について説明する。図41に対応する図45に示す如く、水中或いは砂中にある脚が蹴り上げられるとする。このような動作の場合、チェックパーツの基本スピードを(PX-0X, Py-0y, PZ-0Z)とする。このベクトルの跳ね上げ角度(COS  $\theta$ )は、図に示すように、ベクトルの内積を使用して、図46に示すように計算される。

ここで、COS  $\theta$ を求める理由は、跳ね上げの角度が大きいほど( $90^\circ$ )に近い程跳ね上げ量を減らすためである。すなわち、水、砂のセット数は、基本セット数・移動量・COS  $\theta$ となる。

砂、水滴のスピードは基本スピードに乱数(0~255)/2550の数値を乗じた値となる。このことは、先の着地発生の場合においても同様である。乱数を乗じることによって、ランダムに発生する砂、水の飛び散りを多彩に表示することが可能となる。

このスピードにさらに(0X, 0y, 0Z)を加算した点が砂、水の飛び散り発生点となる。図47を参照すると、乱数が1/10(=255/2550)の場合、発生点は図47に示す座標によって表示される。なお、砂、水が図40に示すように順次パターンチェンジ(32パターン)される。

既述の継続処理は、このようにして決められた、着地発生点、跳ね上げ発生点を使用して、砂、水の飛び散りが継続的に表示される。

図48は、以上説明したこの実施形態（飛翔体運動処理）のフローチャートである。S1600において、チェックパーツの下が水面か否かが判定される。この判定は、水  
5 或いは砂の跳ね上げが予定されているステージ（シーン）について行われる。換言すれば、闘士が立っている地面が岩場、土、或いはその他の固形物であるようなステージでは、この判定を行わない。こうすることにより、様々なステージが存在しても、このような飛翔体の処理を必要なステージにのみ迅速に行うことができる。

パーツの下が地面ではなく水である場合は、次のステップに移行する。このステージ  
10 では既述の設置面属性フラグ(b water)が「1」に設定されている。砂上のステージである場合は、砂地面の処理（S1602）に移行する。なお、この処理は、水面上の処理と同じであるので、その説明を省略することとする。

次の処理、S1604においては、キャラクタ（闘士）の各パーツが着地（着水）しているか否かが、キャラクタの今回ポジションのY座標(Py)と水面位置のY座標(epos)  
15 とを比較することによって判定される。この判定によって、チェックパーツが水面下に在る場合は、着地発生処理に移行し、これとは反対の判定の場合は、跳ね上げ発生処理に移行する。

S1606では、前回のインターラプトにおいて着地状態であったか否かが着地判定フラグをチェックすることによって判定される。このフラグ(b air)が「0」である場  
20 合は、数インターラプトの間のこの着地状態が継続される（S1608）。一方、このフラグが「1」である場合は、今回のインターラプトにおいて新たに着地が発生したとして着地発生処理（S1610）をするとともに、このフラグを「0」にリセット（S1612）して図34のメインルーチンにリターンする。

一方、チェックパーツのY座標が水面位置より高い場合は、S1614に移行して前  
25 回のインターラプトにおいて、着地状態であったか否かが着地判定フラグをチェックすることによって実行される。前回着地状態であったとき、すなわち、チェックパーツが水面下にあった場合は、例えば、今回図に示すように脚の蹴り上げがあったとして、水

を脚の動きに合わせて上方に跳ね上がるパターンが表示される（S 1 6 1 6）。そして、このフラグを「1」にセットして（S 1 6 1 7）リターンする。一方、前回着地状態にないときは、跳ね上げの処理が継続される（S 1 6 1 8）。

以上説明したように、この実施形態によれば、キャラクタの移動量、移動方向を、キャラクタの動きに合わせて移動する、設置物（水、砂等）に反映させることが可能となる。

次に、ゲーム中のキャラクタ（闘士）の運動によって風を発生させ、この風によってステージに落ちている葉（飛翔体の一つ）を舞い上げる運動（飛翔体運動処理の一種）をシミュレートとする処理について説明する。

図 4 9 を参照すると、1 7 0 は運動する闘士を示し、1 7 2 は運動する闘士によって舞い上げられた葉を示す。ここで云うところの「風」とは、数インターラプト（インターラプトとは、一コマの表示のための期間のこと、この場合は 1 / 6 0 秒、換言すれば、垂直同期信号同士の間）の持続するものではなく、1 インターラプト内に葉の挙動に影響を与えるベクトルのことを云う。

この処理は、図 5 0 に示す概略フローチャートによって実行される。このフローチャートは、キャラクタの運動による風が発生するか否かのチェックを行う（S 1 8 0 0, S 1 8 0 2）。次いで、風が発生することをシミュレートする場合は、葉の舞い上がりをチェックするルーチンに移行し（S 1 8 0 4）、次いで、葉の運動をシミュレートする（S 1 8 0 6）。前記ゲーム機は、最初に葉を空中に舞い上げてこれを落下させる状態のシミュレートし、この葉が地面に到達する迄はこの自由落下運動を継続し、この葉が空中から地面に到達した以降は、葉が停止する迄地面上を這う運動をシミュレートする。風が発生しない場合には、葉の舞い上がりチェックを行うことなく、葉の運動を継続する（S 1 8 0 6）。

25 25 先ず、風の発生チェックのためのフローチャートを図 5 1 に基づいて説明する。S 1 9 0 0 では、パーツポジションを計算する。この計算は、闘士が一对存在する場合には、闘士 1, 2 に対して交互に 1 インターラプト毎に行われる。チェックするパーツは、右

脚、左脚、頭の合計3箇所について行われる。それぞれのパーツの移動量をチェックして風の発生の有無が判定される。この判定は、次のとおりである。

図に示すように、チェックするパーツの前回（1インターラプト前）のポジションを( $Ox, Oy, Oz$ )とし、今回のポジションを( $Px, Py, Pz$ )とする。この時の移動量 $M$ は、図51の(1)式に示すとおりである。ここでは、 $XZ$ 座標系での移動量 $M'$ （式(2)）を使用する。

S1902において、 $M'$ が0.2より大きい場合は、パーツの移動量が風を発生させる程度に大きいとして、次のステップに移行する。 $M'$ がこの値より小さい場合は、風が発生しないものとして、このルーチンを終了する。

10 S1904では、 $Y$ 方向の変化量( $Py-Oy$ )が、( $Py-Oy$ ) $\leq -0.8$ である場合、風の向きが極端に下方であるとして風を発生させないものとする。S1906では、風の発生フラグを「風発生あり(=1)」に立てて、かつ風の発生ポジション、および風の大きさ(ベクトル)を決定する。

風の発生ポジション及びこの $X, Y, Z$ 方向のベクトルは、図に示すとおりである。

15  $M'$ は、 $0.12 \leq M' \leq 1.0$ に修正される。この $M'$ を風の $Y$ 方向のベクトルとする。 $X, Z$ 方向のベクトルは、 $X$ 方向、 $Y$ 方向の変化量に0.6を乗じた値とする。ここで、0.6を乗じているのは、 $X, Z$ 方向のベクトルが $Y$ 方向のベクトルより相対的に小さい値として、主に風がモデル空間の上方に舞い上がる状態をシミュレートするためである。

20 風は、一つのインターラプトの間一つベクトルしか発生させないものとする。複数箇所において、風が発生した場合には、その中でもっとも値が大きいものを発生すべき風とする。

次に、葉の舞い上がりチェックのフローチャートについて説明する(図52)。葉の一つ一つについて、風発生ポジションと葉のポジションとから葉が風の影響を受けるか否かをチェックして、この影響範囲内であれば、風の影響を受けて葉の運動(スピード、ベクトル)が変化する。

ステップ2000では、風のデータと葉のデータが読み込まれる。これらのデータ( $X$

成分、Y成分、Z成分)は、図に示すとおりである。S 2 0 0 2では葉の高さに関する演算を行う。この演算は次のとおりである。地面の高さを  $epos$  とすると、地上1 mの高さは  $epos + 1.0$  (m) となる。この高さから葉の高さ

1  $ypos$  を引いた値を  $C 1$  とする。

5  $C 1 = epos + 1.0 - ypos$  (m)

$C 1 \leq 0$  のときは、葉の位置(地上から1 m以上にある。)がキャラクタの動作によって発生した風の影響を受け無い程度に高いと判定されて、この処理を終了する。このとき葉は風の影響を受けない(影響度0%)。一方、風の影響があると判定された時(S 2 0 0 4 : No)、すなわち、 $C 1 > 0$  の時は、 $C 1$  の値に応じた影響係数を決定する

10 (S 2 0 0 6)。例えば、葉が地上にあるとき

( $ypos = epos$ ,  $C 1 = 1$ ) は、葉は風の影響の100%の影響を受ける。 $C 1 = 0.5$  の時は50%の影響を受ける。 $C 1$  は、 $0 \leq C 1 \leq 1.0$  の値をとる。これらのことは、図68に説明されている。この影響係数は、葉の高さによって、葉が風の何%の影響を受けるかということ指標するためのものである。

15 次のS 2 0 0 8は、風の発生ポジションと葉のXZ平面における距離 $L'$ を演算する。図に示す $L'$ の計算式中の  $w yspd$  は、既述のとおり風のY軸方向のベクトルであり、且つ、葉に対する影響距離(半径)も兼ねている。

$w yspd - L'$  が0以下のときは、葉は風による影響範囲外として、風による影響は無いものとしてリターンする(S 2 0 1 0)。一方、0を越える場合は、

20  $C 2$  (距離による影響係数)を算出する。さらに、この $C 2$ を修正して $C 2'$ を算出する。この影響係数の例が図68に示されている。

S 2 0 1 4では、葉に対して風の影響(ベクトル: X方向, Y方向, Z方向)、すなわち、葉のスピード(ベクトル)を図に示すように決定し、葉の現在ポジションにこの影響を加えて葉の運動をシミュレートする(図50のS 1 8 0 6)。葉の運動の処理ルーチンは、このすぐ後に説明される。葉のX方向, Z方向のスピードは、常に新しく設定され、Yスピードは前回(前のインターラプト)のスピードに加算される。

次に葉の運動をシミュレートする詳細フローチャートについて説明する(図53)。



葉の運動は、図 5 4 に示すように、葉の中心ポジションの運動と、葉のモデルの X, Y rot の回転運動の二つの形態のものが与えられている。

(1) は風の影響や重力によって葉の中心ポジションが移動する運動を示したものであり、(2) は、葉が「ひらひら」と舞う運動を X rot, Y rot によってシミュレート（表現）することを示している。これらの運動は、概略フローチャートで説明した、自由落下運動の一環である。

前者の運動（空中にある場合、そして空中から着地した場合）の特性式は、図の S 2 1 0 0 に示すとおりである。葉が空中にある場合、X 方向の速度

(l xspd) に 0. 8 7 を乗じて新たな X 方向の速度としている理由は、空気抵抗を考慮に入れたためである。Z 方向についても同様である。Y 方向の速度は、0. 0 0 2 を減じて新たな Y 方向の速度としている。これにより、葉の重力方向の速度が徐々に増して行くことがシミュレートされる。但し、Y 方向の落下速度 (l yspd) の最大値は - 0. 0 0 6 とする。葉が着地している場合については、後述する。

一方、図 5 5 に示すように、葉 2 3 0 がひらひらと舞う状態は、既述の X rot と Y rot とによってシミュレートされる (S 2 1 0 2)。但し、この実施の形態では、3 2 パターンの X rot を用意して、順次パターンチェンジしている。Y rot による回転は、次のようである。

Y 回転速度 = l yspd × 1200 × 256

1 回転 (360°) を 0 × 0000 ~ 0 × FFFF (0 ~ 65536) で表現している。1° ■ 182 (0xB6) である。なお、Y rot もパターンチェンジにすると、葉の動きが粗くシミュレートされるおそれがあるので、一方の回転要素 (X rot) をパターンチェンジとし、残りの要素 (Y rot) を計算によって求めている。

葉が地面に到達した後の運動は、図 5 3 に示すように、中心運動特性式と回転運動特性式によってシミュレートされる (S 2 1 0 0, S 2 1 0 2)。中心運動特性式は、図 5 3 に示されている。この運動特性式では、葉の Y 軸方向の速度

(l yspd) は 0 に設定されているので、地面上を葉が回転しながら動く状態がシミュレートされる。この特性式において、X 方向の速度と Y 方向の速度のそれぞれに 0. 5 を

乗じているのは、地面の摩擦を考えてのことであり、空中に葉がある場合よりも小さい値を乗じて空気抵抗よりも地面の摩擦抵抗が大きいことをシミュレートしている。尚、X軸周りの葉の回転角度(X rot)が「0」であるか否かが判定され「0」である場合は、葉が地面に対して平行に接したとして葉の動きを停止させる。したがって、キャラクターが動くことによって発生した風によって葉が空気中に舞った後、この葉が地面に落下して地面と平行になった時点で葉の動きが停止される。

空中に舞った葉が着地したか否かは、地面のY座標と葉のY座標を比較して判定される。葉のY座標 $\leq$ (又は「=」)地面のY座標であるときに、葉は着地状態にあると判定する。なお、葉は一つのポリゴンに葉の模様(テクスチャー)を貼り付ける(マッピング)することによって形状モデリングされる。

以上説明した、飛翔体(水、砂、葉)は、その移動量や移動方向が、キャラクター(モデル)の動作の特性によって決まるために、モデルの動きを飛翔体に正確に反映させることができる。このような飛翔体は本来モデルの動きによって飛んだり、或いは落下したりするものであり、モデルの動きをこれらに反映させることにより飛翔体の動作を高品質にシミュレートできる。また、以上説明した飛翔体の運動処理によれば、仮想空間内におけるモデル(飛翔体)の(自由)落下運動を演出感豊かに、且つ高品質に再現できる。例えば、葉が「ひらひら」と舞う映像が生成されていることである。

#### 自由落下運動処理

次に、仮想空間内を落下する対象物の運動処理について説明する。ここでの実施例は、対象物を雪として、雪に「ゆらぎ」運動を与える演出をシミュレートすることについて説明する。ここで説明する「揺らぎ」とは、雪が風によって舞いながら降っている状態、或いは関士の動きによって舞い上がった落ち葉が舞い落ちる状態等比較的不規則な運動を云う。この実施例は、この「揺らぎ」の状態を効率良くシミュレートするようにしている。

そこで、雪が舞う実施例について説明する。舞い落ちる雪一つは一つ又は複数のポリゴンによって構成されており、モデル空間に設定されている風のベクトル(このベクトル

ルは適宜変化させている)の影響を受けている。雪のポリゴンの運動は、重力による自由落下、風ベクトルによる影響(以上、請求項の第1の手段)、仮想空間のXZ平面(水平面、すなわち地表面に平行な座標系)の円運動(同項の第2の手段)によってシミュレートされる。円運動の角速度、振り幅は、メモリ内にテーブル化されており、その一つが選択されて使用される。

雪の挙動に揺らぎを与える、既述の円運動は、次のようにして与えられる。角速度を $\omega$ とし、回転角を $q$ とし、振り幅を $a$ とすると、揺らぎ運動のX成分( $x\ off$ )とZ成分( $z\ off$ )、すなわち雪の基本挙動(自由落下と風によるベクトルによって決定される。)に対するオフセット量は、次のとおりである。

$$\begin{aligned} x\ off &= \sin(q + \omega) \cdot a \\ y\ off &= \cos(q + \omega) \cdot a \end{aligned}$$

このオフセットデータは、毎インターラプトの間10回計算されて、雪の基本ポジションに加算される。

雪の運動方程式は、次のとおりである。

$$\begin{aligned} x\ pos &: \text{雪のポジション (X方向)、} \\ y\ pos &: \text{雪のポジション (Y方向)、} \\ z\ pos &: \text{雪のポジション (Z方向) とする。} \\ x\ pos &= x\ pos + x\ off + wind\ x \text{ (風ベクトルスピードの x 成分)} \\ y\ pos &= y\ pos + fall \text{ (雪の落下速度、例 - 1 cm / int)} \\ z\ pos &= z\ pos + z\ off + wind\ z \text{ (風ベクトルスピードの z 成分)} \end{aligned}$$

本実施例によれば、雪が風に揺られて舞い落ちる挙動(図56参照)が、前記運動方程式によって表現される。したがって、複雑な雪の「揺らぎ」運動が簡単な計算式によって得ることができ、CPUブロックに与える計算負荷も大きなものには至らない。

図56はこの雪の移動軌跡を示すものであり、所定方向に大きさや方向が変わるベクトルを自由落下する雪に加えても「揺らぎ」の状態を的確にシミュレートすることはできない。そこで、この雪に円運動(周回運動の一種、その他楕円運動のようなものでも

良い。)を加えることによって、この「揺らぎ」を再現できるようにしている。

この処理によれば、仮想空間内のモデルの自由落下運動をシミュレートする際の品質を高くすることができるために、落下画像処理の演出感を豊かに醸し出すことが可能となる。

#### 5 不定形区画における衝突判定処理

次に不定形の区画とモデルとの衝突処理について説明する。この処理は次のような内容を有している。ここで、区画とは闘士の動きを規制する領域であり、例えば、周囲が不定形の壁で囲まれたリングを想像すると理解し易い。闘士はこのリング内を自由に移動できるが、壁を越えてリング外に移動することはできない。この壁がここで説明する

10 「区画」に相当する。

図57の(A)には不定形の区画250が示されている。この区画は仮想空間のXZ平面(Y軸は高さ方向)に設定されている。この区画は7角形として設定されている。この区画としては様々な形態のものを設定可能である。このような区画としては、周囲が岩場であるリングを想定した場合のその岩場を云う。

15 図57はこの処理の原理をも示している。壁(区画)が実際に存在する様に見せるためには、キャラクタが壁を越えて移動しないよう、キャラクタの位置を補正する必要がある。この実施形態は、大きさや形を任意に設定することができる闘技場の壁とキャラクタとの当たり判定を行い、位置を補正するための壁の法線方向へ追い出しベクトルを算出する。

20 図57の(A)は、この衝突判定のモデル図であり、(B)はベクトル計算の原理図であり、(C)はキャラクタ(モデル)と区画との衝突判定モデル図の一部拡大図である。なお、図57にはベクトル計算式も同時に記載されている。

(A)に示すように、この当たり判定に際して、キャラクタの形状を球によって近似し、さらに壁及びこの球体を地面と平行な平面(XZ平面)に平行投影して、区画の領域を形成する各辺と、球が投影された結果得られた円252との位置関係を調べる。こ  
25 ここで、球の位置はキャラクタの計算上の位置である。この図において、Lはこの区画の任意の辺の一頂点であり、Rはこの辺の他の頂点である。ある任意の辺について、反時

計方向にLが来るように設定される。

図中のTはキャラクタが投影された円の中心点である。ベクトルR、ベクトルT、ベクトルLは、この座標系のある任意の一点からこれらの各点に対して設定されたベクトルである。また、ベクトルPは円の中心座標から区画の各辺に対して垂直に引かれたベクトルである。ベクトルVはこの円に対する辺L Rに平行な接線と円との交点から、辺L Rに対して直角に引かれたベクトルであり、この処理における追い出しベクトルに相当する。

この処理においては、追い出しベクトルVが算出されることが目的である。円の中心が直線R Lの外側（これを正領域とする。）にある場合は、図に示すように、追い出しベクトルは図の式（1）に示されるように設定される。

一方、円の中心が直線R Lの内側（これを負領域とする。）にある場合は、式（2）のように設定される。すなわち、円の中心と直線R Lとの間の距離が、半径（d）以上の場合は、追い出しベクトルは設定されない。この距離が越える場合は、追い出しベクトルが設定される。このとき、ベクトルは壁の方向に向かっているために、単位ベクトルP eを円の中心が直線R Lの外側にある場合とは反対方向に設定される。

図5 8はこの実施例の動作について説明するフローチャートである。まず、コントローラのボタン或いは方向キーの操作量が読み込まれる（S 2 6 0 0）。この操作量から、計算上のキャラクタの位置が決定される（S 2 6 0 2）。すなわち、区画が無いとした場合のキャラクタの計算上の位置であって、コントローラの操作によってこの位置が要求される。

次いで、S 2 6 0 4において、このキャラクタの位置とある辺R Lとの比較が行われる。すなわち、図5 7の式（3）の関係がチェックされる。この判定が否定された場合（ $\sim 1$ ）は、次の辺R Lについてこの関係をチェックする。この場合は、点Tからこの辺に直角にベクトルPを引くことができないとし（すなわち、衝突判定の対象となる辺とはされない。）、S 2 6 0 6のベクトルPの計算処理をすることなくS 2 6 0 8に移行する。

この関係が肯定された場合は、S 2 6 0 6にて図5 7に示すところによって、ベクト

ルPの値を算出する。次に、S 2 6 0 8にて全ての辺について、S 2 6 0 4の処理が終了したか判定され、否定された場合はS 2 6 1 0に移行して他の辺についての比較が実行される（S 2 6 0 4）。一方、これが肯定された場合は、S 2 6 1 2に移行し、ベクトルPの最小値が選択される。このベクトルPが設定された辺が追い出しベクトルを法線方向に設定するための壁（辺、区画）として決定される。すなわち、このベクトルが設定される辺の方にキャラクタが接触していると判定される。

そして、次のステップ（S 2 6 1 4）では、円の中心が区画内にあるか否かが判定され、区画外のときは図の式（1）によって追い出しベクトルが設定され、区画内の上には式（2）によって追い出しベクトルが設定される。式（1）の場合及び（2）において、ベクトルPの大きさが円の半径より小さい場合は、キャラクタと壁との間に当たり（衝突）があるとされて、既述の追い出しベクトルが設定される。この追い出しベクトルによって、実際の映像では、キャラクタがこの区画（壁）から先に移動することが規制される画像処理が実行される（S 2 6 1 6）。

もっとも、この壁から追い出しベクトルによってモデルが壁から辺の法線方向に定義されたベクトルVによって弾かれるような画像処理を実行するようにしても良い。図6 7はその時に於ける画像処理の一例であり、符号3 5 0は闘士であり、符号3 5 2は不定形の壁である。闘士3 5 2が図中区画の側（図中左方向）に移動しようとしても、区画に遮られて移動できない状態を示している（図の（1）の状態）。但し、図の（2）に示すように、闘士3 5 0の全体は区画によって移動できない状態であるが、右脚3 5 4（キャラクタの一部のパーツ）が両方向の矢印に示されるように進退するようにすることはできる。このことにより、遊戯者は自己が操るモデルが区画に遮られて、これ以上その方向に移動できないことを示ることができる。

この実施形態に示される処理によれば、各辺毎に追い出しベクトルを算出するようにしているために、不定形の区画であってもこの区画とキャラクタとの当たり判定を確実に行うことができ（すなわち、追い出しベクトルの値が与えられた時には当たり判定が肯定される。）、この判定結果を画像処理に反映することが可能となる。

テクスチャー列の作成

次に繰り返すテクスチャー列を作成する形態について説明する。図59はこのことを示す説明図である。下段(Bottom)には、第1のテクスチャー列が示されており、中段(Top)には第2のテクスチャー列が示されている。上段(Togeter)には、第1のテクスチャー列と第2のテクスチャー列とを重ねた結果の第3のテクスチャー列が示されている。

第3のテクスチャー列は、所定の周期を持って繰り返す映像「海上の反射パターン」の映像であることを目的に形成されたものである。第3のテクスチャー列は、0～29の30カット(枚)の絵から構成されており、これらの絵を順次、ポリゴンにマッピング、すなわち、0～29番までの絵をマッピングした後、さらに0～29番までマッピングし、これらを繰り返すことにより、これらのテクスチャー列のつなぎ部分(例えば28～2番)での絵の変化が自然的(連続的、換言すれば、絵飛びが無い)であるようにしている。これにより、海面上の反射パターンが繰り返す映像を作成できる。

ここで、第1、2のテクスチャー列は、既述の市販アプリケーションソフトウェアを利用して作成される。テクスチャー列の発生に当たってパラメータの値はそれぞれの列に応じて異なるようにしている。但し、ソフトウェアの制約から、それぞれの列において同じであるが、テクスチャー列の最初の領域と最後領域のテクスチャー群は、つながりの悪い絵柄になっている。すなわち、例えば、第1のテクスチャー列の0番と29番、そして第2のテクスチャー列の0番と59番である。

そこで、この処理においては、第2のテクスチャーに順次変化する透明度のパラメータ(0～100%)を与える。第2のテクスチャー列の上に記載されている数値がこの透明度に関する値である。100%とあるのは、透明度が全く無い状態を示し、他に7%とあるのは、透明度がより進んだ状態(透明度93%)であることを示している。全く透明にする場合には、0%が与えられる。第1のテクスチャーは、100%のパラメータを与えている。

第3のテクスチャー列は、第1のテクスチャー列と第2のテクスチャー列を図示するように順番に重ねられて作成される。すなわち、第1のテクスチャー列の0番と第2のテクスチャー列の30番が重ねられて第3のテクスチャーの0番が形成され、第1のテ

クスチャー列の1番と第2のテクスチャー列の31番が重ねられて第3のテクスチャーの1番が形成され、・・・、第1のテクスチャー列の29番と第2のテクスチャー列の59番が重ねられて第3のテクスチャーの29番が形成される。

このとき、第1のテクスチャー列の0番と第2のテクスチャー列の30番が重ねられるとき、第2のテクスチャー列の30番の透明度は0であるために（全く透明でない状態）、第1のテクスチャー列の0番は第2のテクスチャー列の30番に全く隠されて、第3のテクスチャー列の0番の映像は、第2のテクスチャー列の30番の映像に等しくなる。

一方、第1のテクスチャー列の29番と第2のテクスチャー列の59番が重ねられるとき、第2のテクスチャー列の30番の透明度は97%であるために（ほぼ透明に近い状態）、第3のテクスチャー列の29番の映像は、第1のテクスチャー列の29番の映像にほぼ等しくなる（換言すれば、第2のテクスチャー列の59番の映像は殆ど見えない状態）。

ここで、第2のテクスチャー列を発生させるとき、パラメータの与え方は、その30番の絵が第1のテクスチャー列の29番の絵に続くような絵となるようにしているために、第3のテクスチャー列を見た場合には、テクスチャー列のつなぎ部分で（図59では28番～2番）において、つながりが既述のように自然となる絵を発生させることができる。

第1のテクスチャー列の終わり部分（例えば25番乃至29番）、第2のテクスチャー列の最初の部分（例えば、30番乃至34番）の絵がつながるようにそれぞれのテクスチャー列にパラメータを与えることによって（あるいは結果的にこうなることによって）、図の矢示で示すように、透明度も勘案すると、第1のテクスチャー列の29番から第2のテクスチャー列の30番につながるようになることにより、第3のテクスチャー列が図示するように自然につながる形態の絵となる。

第3のテクスチャーは、メモリの所定領域に記憶されており、海面を示すポリゴンに順次マッピングされて、海面の白波（画面上では輝度が高い部分、図27では白く表示されている部分）の模様や形態が周期的に繰り返す映像を表現することができる。図示



する例では、1 / 30 秒毎に一枚の絵が表示されるとし、30枚の絵であるとする、1秒毎に海面の白波部分の模様が周期的に繰り返す映像を再生することができる。

波の形態を再生する速度、及び使用するテクスチャー数は、適宜変更可能である。画面の中心に闘士が立っているステージがあるとする、ステージ近傍（すなわち、画面  
5 の中心付近）では、既述の再生速度（一コマ当たり1 / 30 秒、30コマ再生）とし、これから遠くの波については、コンピュータの計算負荷を低減する観点から、一コマ当たりの再生速度を下げ、かつ合計の使用コマ数を少なくすることができる。例えば、第3のテクスチャー列の0番、10番、20番、29番のコマを使用する。このようにしたのは、画面の周辺付近にある映像の再生を多少粗くしても、遊戯者に与える悪印象は  
10 少なく、かつ、コンピュータグラフィックス装置の計算負荷を低減する事の方が有利であるからである。

既述のゲーム装置のメモリには、第3のテクスチャー列が記憶されている。したがって、コンピュータグラフィックスを用いたゲーム機分野に、既述のキャラクター列を利用することができる。ここで、説明したキャラクター列の発生方法は、コンピュータ  
15 グラフィックスの技術分野ばかりでなく、アニメーションの動画作成の技術にも利用することができる。

#### スロー再生処理

図61乃至64は、スロー再生処理（再生速度変更処理）の原理を説明するためのキャラクターの動作図である。二つの闘士290、292が互いに向き合って、両者の闘技の  
20 優劣を競いあっている状態を示している。図61において、290は攻撃を受けている闘士であり、292は攻撃側の闘士である。符号294が闘士290が闘士292から闘技（攻撃：パンチ）を受けた攻撃部位である。図61と図61を上方より見た平面図である図62は闘士290が攻撃をそのまま受けている状態を示している。

一方、図63とその平面図である図64に示すように、闘士290が闘士292から  
25 の技を右手のパーツで払い（すなわち、防御）、それが成功すると闘士292が矢示の方向に仰け反る（ずれる）ようなモーションが再生される。このとき、闘士292のモーションの再生速度を低下させてスロー再生を行う。

このように、一方のキャラクタ（闘士）が他の闘士に技（「パンチ」）を繰り出したとする。一方、他の闘士がこのパンチを払う技をほぼ同時に出したとする。なお、これらのパンチや或いは払い技は、遊戯者がコントロールパッドの操作ボタンや方向キーを適宜操作することによって発生する。

- 5      このパンチの払いが成立したときに、攻撃側の闘士が後方に大きく仰け反るモーションが再生される。このとき、このモーションの再生速度を防御側の闘士の映像の再生速度よりも小さくなるようにする。すると、このスロー再生により、攻撃側の闘士にいわゆる「隙」が生じて、防御側の闘士から攻撃技を攻撃側の闘士に繰り出し易くなる。闘士292は一旦仰け反った後、図65のように元の体型（仰け反る前の）に徐々に復帰する。
- 10      する。

図66は、この処理を示すフローチャートである。S3400において、防御側の闘士が既述の払い技を出す。S3402では相手側の攻撃技（図の場合は「パンチ」）が払い技に当たったか（すなわち、払い技の動作コマンド下にある手のパーツと攻撃側の闘士の手のパーツの衝突判定）の判定を行う。

- 15      この判定が否定された場合は、スロー再生の処理を行うことなくリターンする。一方、この判定が肯定された場合は、攻撃側のキャラクタが防御側のキャラクタの正面に位置しない場合にはリターンし、正面の場合には、S3406に移行する（S3404）。S3406では、攻撃技が払うことができる技であるか否かを判定する。例えば、パンチであれば払うことができ、キックであれば払うことができない等である。この判定は、
- 20      技毎の専用フラグを設定してフラグレジスタの内容を参照することにより容易に実現される。

- 払い不可能技である場合は処理を終了し、払い可能技である場合はS3408に移行する。このステップでは、スロー再生されるべきコマ数を設定し、これをタイマーに保存する。次のステップS3410では、技によって払われる量（図の例では、攻撃側の闘士が仰け反るモーションを再生する場合、各パーツが仰け反る量）を設定する。コマ数及び払い量は、所定の特性式によって計算され、或いはテーブル化されている。
- 25      する。

S3412では払い量を設定されたコマ数で割り、一コマ分のリーフ量（空間座標系

に於ける移動量、及び移動方向)を算出する。そして、S 3 4 1 4 では、攻撃側の闘士の再生速度(表示速度)を防御側の闘士の例えば半分にする。例えば、攻撃側のキャラクターの再生速度を1 / 6 0 秒の半分1 / 3 0 秒とする。

次いで、S 3 4 1 6 では、前記タイマーの値を判定し、「0」である場合は、スロー再生処理は終了されたとして、攻撃側闘士の再生速度を元の状態に復帰させて(S 3 4 1 7)、リターンする。一方、「0」でない場合は、こま当たりのリーフ量にタイマーの値を乗じて攻撃側の移動量とする。そして、S 3 4 2 0 では、この移動量を攻撃側の闘士の座標値に加算する。すなわち、図5 1 の攻撃側の闘士2 9 2 の位置に、図6 3 の払い技が成立した場合の移動量(仰け反り量: 本来のポジション(図6 1 の闘士2 9 2 の位置))を加える。

次いで、S 3 4 2 2 ではタイマーの値(T)を1 減じて、S 3 4 2 4 では、攻撃側の闘士2 9 2 (図6 3) のモーション位置を再計算することが行われる。そして、メインルーチンにリターンする。

この処理によれば、払い技が成功したときに、攻撃側の闘士が最初大きく仰け反りながら、スロー再生状態でさらに徐々に仰け反って行く映像が再生される。このとき、防御側の闘士は攻撃技を攻撃側の闘士に対して繰り出し易くなる。

この処理によれば、一方の闘士のモーションの再生速度を低くしながら他方の闘士のモーションの再生を行うことができるために、遊戯者は自己が操る防御側の闘士技をより有効に繰り出すことができることになり、このことは、ゲームの要素としてのスロー再生を可能にしたことに通づることになる。したがって、スロー再生を演出感豊かに表現することが可能となる。

なお、本発明は上述した各種の実施例の構成に限定されるものではなく、当業者であれば、請求の範囲記載の構成の範囲内でさらに様々な変形が可能であろう。

なお、ゲーム機の動作用プログラムが記憶された記憶(記録)媒体としては、既述のカートリッジROM、CD-ROMの他にインターネット、パソコンネット上の通信媒体でも良く、また、電子メールのサーバーをも含むものである。

## 請 求 の 範 囲

1. 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、仮想視点に基づくこの仮想空間の映像を表示手段において表示するようにした画像処理を行うゲーム装置であって、前記モデルに仮想的な向心力を与える画像処理手段を備えることを特徴とするゲーム装置。  
5
2. 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、仮想視点に基づくこの仮想空間の映像を表示手段において表示するようにしたゲーム用画像処理装置であって、前記モデルに仮想的な向心力を与える画像処理手段を備えることを特徴とするゲーム用画像処理装置。  
10
3. 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、仮想視点に基づくこの仮想空間の映像を表示手段において表示するようにしたゲーム用画像処理装置であって、前記モデルが移動している時と静止している時との摩擦量を変えて、このモデルに与える画像処理手段を備えることを特徴とするゲーム用画像処理装置。  
15
4. 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、仮想視点に基づくこの仮想空間の映像を表示手段において表示するようにしたゲーム用画像処理装置であって、前記モデルが置かれるステージの表面形状に合わせて前記モデルの投影像を表示させる画像処理手段を備えることを特徴とするゲーム用画像処理装置。  
20
5. 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、仮想視点に基づくこの仮想空間の映像を表示手段において表示するようにしたゲーム用画像処理装置であって、前記仮想視点が表示対象としての前記モデルに対して成す視野領域と、表示対象として設定されていない他のモデルとの重なり判定を行い、この判定が肯定された場合に、この他のモデルが表示されないような画像処理手段を備えることを特徴とするゲーム用画像処理装置。  
25
6. 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方

向に移動するように制御されるとともに、仮想視点に基づくこの仮想空間の映像を表示手段において表示するようにしたゲーム用画像処理装置であって、前記モデルの予定された移動軌跡の終了点に対して、この終了点とは異なる目標点が設定され、前記終了点をこの目標点に一致させるように移動軌跡が補間される画像処理手段を備えることを特徴とするゲーム用画像処理装置。

7. 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、仮想視点に基づくこの仮想空間の映像を表示手段において表示するようにしたゲーム用画像処理装置であって、前記モデルが前記仮想空間内に置かれている高低差を求め、この高低差に応じてこのモデルに対して与えられている動作を補正する画像処理手段を備えることを特徴とするゲーム用画像処理装置。

8. 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、所定位置の視点に基づくこの仮想空間の映像を表示手段に表示するようにしたゲーム用画像処理装置であって、モデルの移動軌跡を残像として表現するための残像表現処理手段を備え、この処理手段は前記モデルの現在モーション以前のモーションデータをそのまま記憶する記憶手段と、この記憶データを前記モデルの現在のモーションデータとともに表示する表示制御手段とを備えることを特徴とするゲーム用画像処理装置。

9. 請求項8記載の装置において、前記表示制御手段は、前記記憶手段のデータに演出を加えて表示するための演出手段を備えるゲーム用画像処理装置。

10. 請求項9記載の装置において、前記演出手段は、半透明の画像処理を行うゲーム用画像処理装置。

11. 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、所定位置の視点に基づくこの仮想空間の映像を表示手段に表示するようにしたゲーム用画像処理装置であって、前記モデルの移動特性を算出する算出手段と、前記仮想空間内に飛翔体を存在させるとともに、この飛翔

体の一連運動を前記算出手段による算出結果に応じて制御する飛翔体制御手段とを備えることを特徴とするゲーム用画像処理装置。

1 2. 仮想空間内に設定されたモデルの自由落下運動をシミュレートするゲーム用画像処理装置において、前記モデルの全体運動をシミュレートする第1の手段と、前記  
5 モデルに周回運動を付与する第2の手段とを備えることを特徴とするゲーム用画像処理装置。

1 3. 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、所定位置の視点に基づくこの仮想空間の映像を表示手段に表示するようにしたゲーム用画像処理装置において、不定形の区画を前  
10 記仮想空間内に設定する区画設定手段と、この区画とこのモデルとの間に所定のベクトルを設定する手段であり、この区画を越えてこのモデルが移動しようとする場合には、このベクトルをモデルが区画を越えない方向に設定するベクトル設定手段と、このベクトルに応じて前記モデルの移動を制御するモデル移動制御手段と、を備えることを特徴とするゲーム用画像処理装置。

1 4. 所定のパラメータを与えて周期的に変化するテクスチャーを形成する装置において、第1のテクスチャー列を作る第1の手段と、第2のテクスチャー列を作る手段とを備えるとともに、第1のテクスチャー列の最後或いは最初のテクスチャーと、第2のテクスチャー列の最初或いは最後のテクスチャーとが連続する形態になるようにそれぞれの手段にパラメータを与えるとともに、第1のテクスチャー列と第2のテクスチャー列の少なくとも一方の透明度を漸増或いは漸減して両方のテクスチャー列の最初の手テクスチャー同士から順に重ねて目的とするテクスチャー列を形成することを特徴としたテクスチャー生成装置。

1 5. 周期的に変化するテクスチャーを形成する方法において、所定の方向に変化する第1のテクスチャー列を作る第1の工程と、同様に所定方向に変化する第2のテクスチャー列を作る工程とを備えるとともに、第1のテクスチャー列の最後或いは最初のテクスチャーと、第2のテクスチャー列の最初或いは最後のテクスチャーとが連続する形態になるようにするとともに、第1のテクスチャー列と第2のテクスチャー列の少な

くとも一方の透明度を漸増或いは漸減して両方のテクスチャー列の最初のテクスチャー同士から順に重ねて目的とするテクスチャー列を形成することを特徴としたテクスチャー生成方法。

16. 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、所定位置の視点に基づくこの仮想空間の映像を表示手段に表示するようにしたゲーム用画像処理装置において、二つのモデル間に所定の条件が成立したか否かを判定する第1の手段と、前記条件が成立したことをそれぞれのモデルの運動に反映させる第2の手段と、この条件が成立したときに一方のモデルの運動の再生速度を他方のモデルに対して変化させる第3の手段とを備えることを特徴としたゲーム用画像処理装置。

17. 仮想空間内に設定されたモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、仮想視点に基づくこの仮想空間の映像を表示する画像処理を行うゲーム装置に使用する、前記画像処理のプログラムを記録した記録媒体であって、前記プログラムは前記モデルに仮想的な向心力を与える一連の手順を記載した情報を含むことを特徴とする記録媒体。

## 補正書の請求の範囲

[1997年11月10日(10.11.97)国際事務局受理:出願当初の請求の範囲1及び17は補正された;新しい請求の範囲18が加えられた;他の請求の範囲は変更なし。(2頁)]

1. (補正後) 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、仮想視点に基づくこの仮想空間の映像を表示手段において表示するようにした画像処理を行うゲーム装置であって、対峙する複数のモデル間に仮想的な引力を設定した画像処理手段を備えることを特徴とするゲーム装置。

2. 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、仮想視点に基づくこの仮想空間の映像を表示手段において表示するようにしたゲーム用画像処理装置であって、前記モデルに仮想的な向心力を与える画像処理手段を備えることを特徴とするゲーム用画像処理装置。

3. 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、仮想視点に基づくこの仮想空間の映像を表示手段において表示するようにしたゲーム用画像処理装置であって、前記モデルが移動している時と静止している時との摩擦量を変えて、このモデルに与える画像処理手段を備えることを特徴とするゲーム用画像処理装置。

4. 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、仮想視点に基づくこの仮想空間の映像を表示手段において表示するようにしたゲーム用画像処理装置であって、前記モデルが置かれるステージの表面形状に合わせて前記モデルの投影像を表示させる画像処理手段を備えることを特徴とするゲーム用画像処理装置。

5. 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、仮想視点に基づくこの仮想空間の映像を表示手段において表示するようにしたゲーム用画像処理装置であって、前記仮想視点が表示対象としての前記モデルに対して成す視野領域と、表示対象として設定されていない他のモデルとの重なり判定を行い、この判定が肯定された場合に、この他のモデルが表示されないような画像処理手段を備えることを特徴とするゲーム用画像処理装置。

6. 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方



くとも一方の透明度を漸増或いは漸減して両方のテクスチャー列の最初のテクスチャー同士から順に重ねて目的とするテクスチャー列を形成することを特徴としたテクスチャー生成方法。

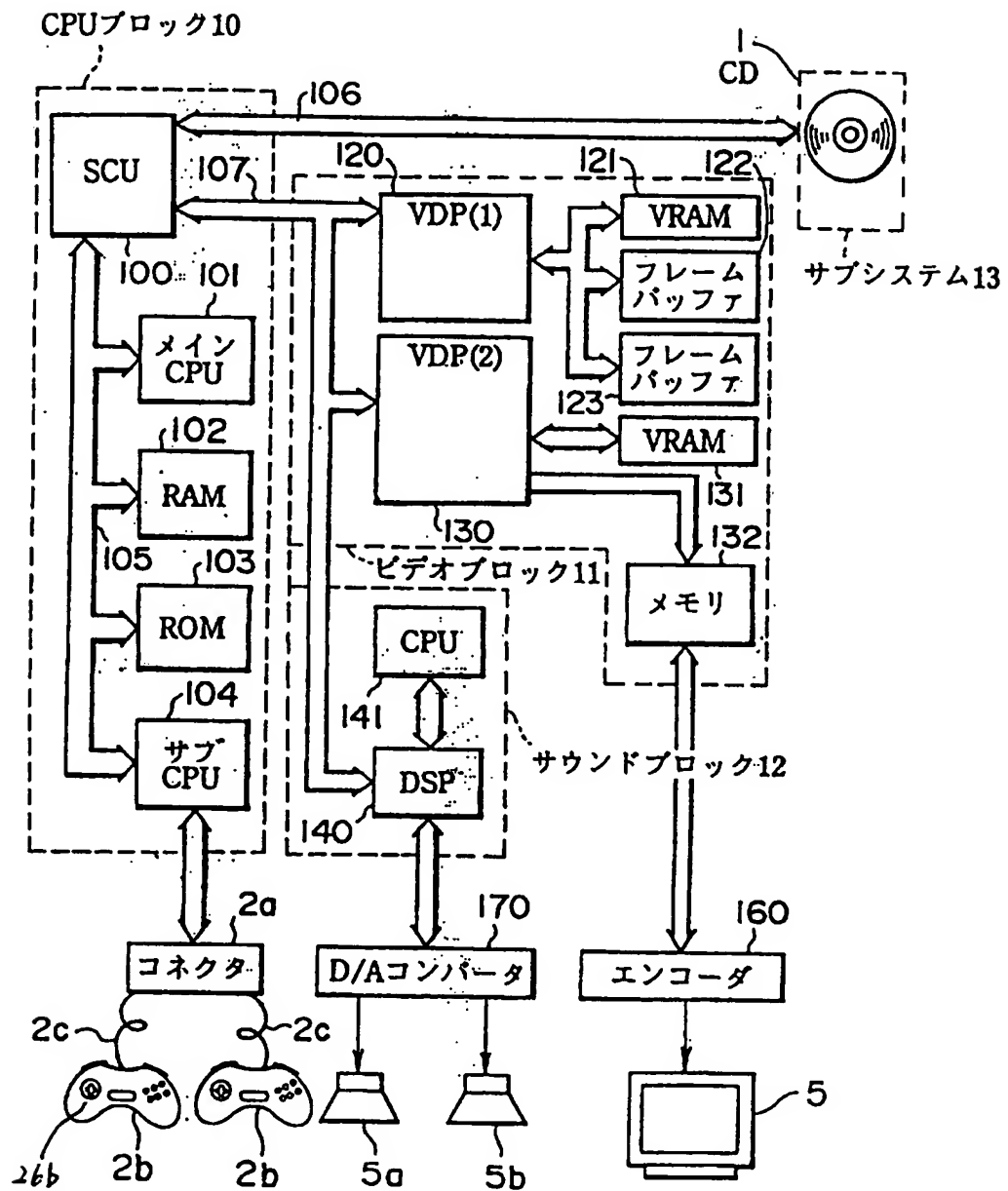
16. 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、所定位置の視点に基づくこの仮想空間の映像を表示手段に表示するようにしたゲーム用画像処理装置において、二つのモデル間に所定の条件が成立したか否かを判定する第1の手段と、前記条件が成立したことをそれぞれのモデルの運動に反映させる第2の手段と、この条件が成立したときに一方のモデルの運動の再生速度を他方のモデルに対して変化させる第3の手段とを備えることを特徴としたゲーム用画像処理装置。

17. (補正後) 仮想空間内に設定されたモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、仮想視点に基づくこの仮想空間の映像を表示する画像処理を行うゲーム装置に使用する、前記画像処理のプログラムを記録した記録媒体であって、対峙する複数のモデル間に仮想的な引力を設定した画像処理を行うための記録媒体。

18. (追加) 仮想空間内に所定数のモデルを設定し、このモデルが前記仮想空間内で所定方向に移動するように制御されるとともに、仮想視点に基づくこの仮想空間の映像を表示手段において表示するようにした画像処理を行うゲーム装置であって、前記モデルの動きを制御するための入力手段と、この入力手段に設けられた、当該モデルの上下左右への動作方向を指定する方向キーと、対峙する複数のモデル間に仮想的な引力を設定する画像処理手段と、を備え、この画像処理手段は前記方向キーの操作によって、前記モデルが自動的に円運動をするように動作させる画像処理を実行するゲーム装置。

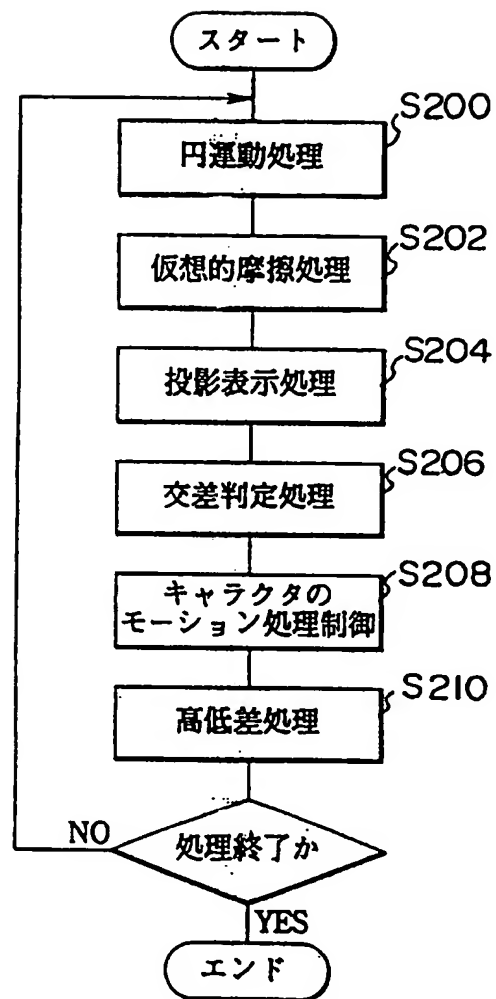
1/60

【図1】



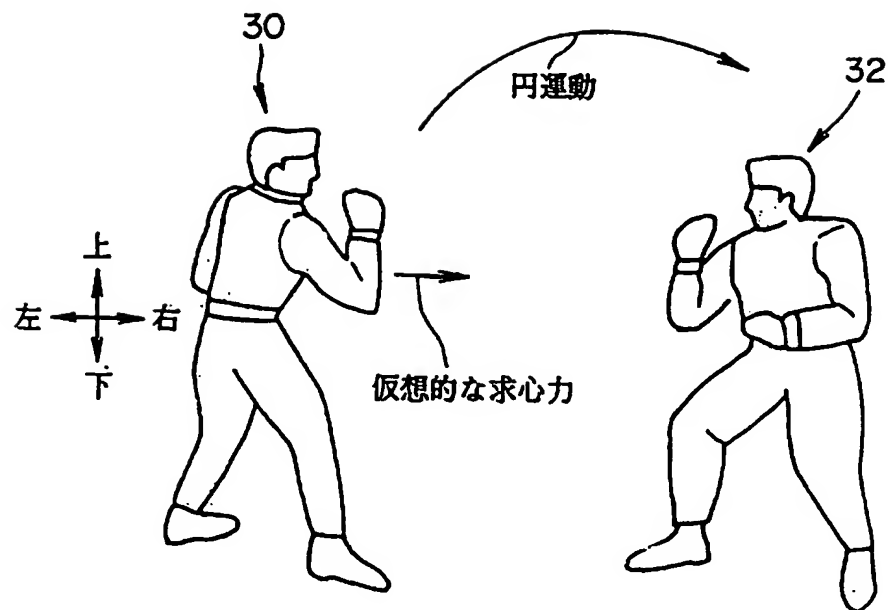
$\frac{2}{60}$ 

【図 2】



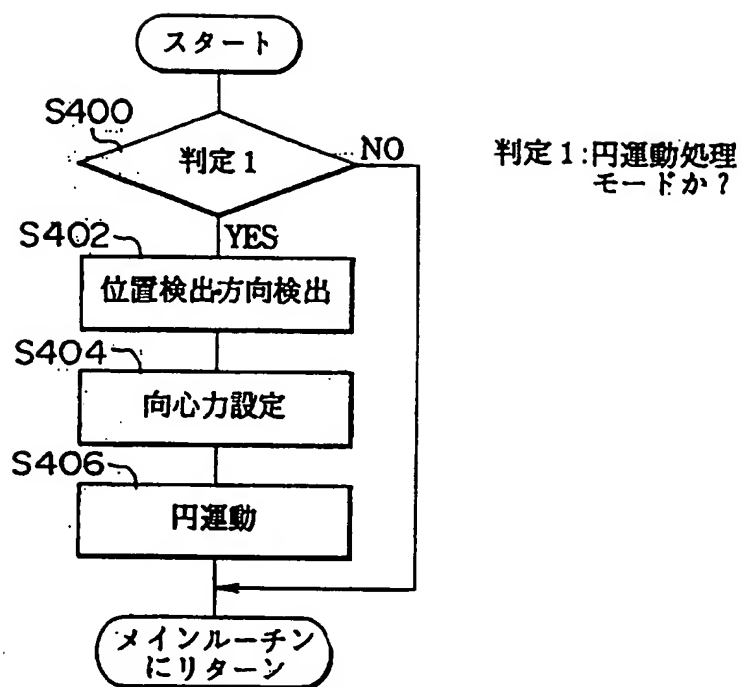
3/60

【図3】



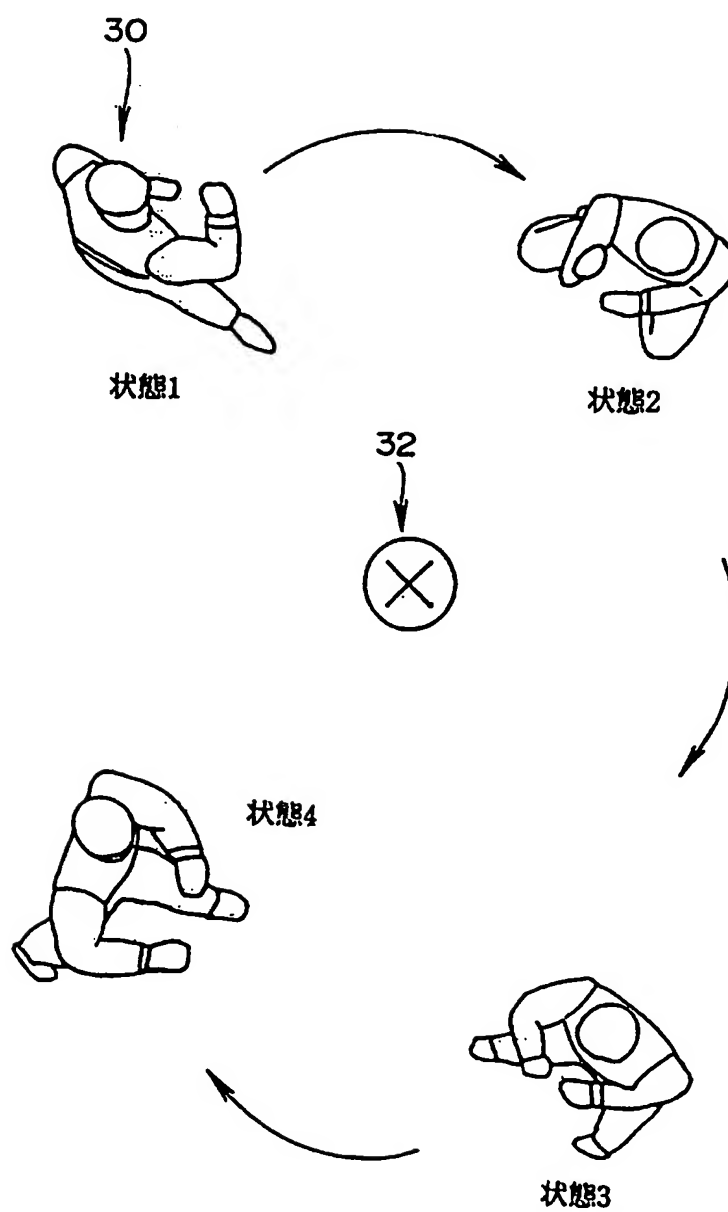
4/  
60

【図 4】



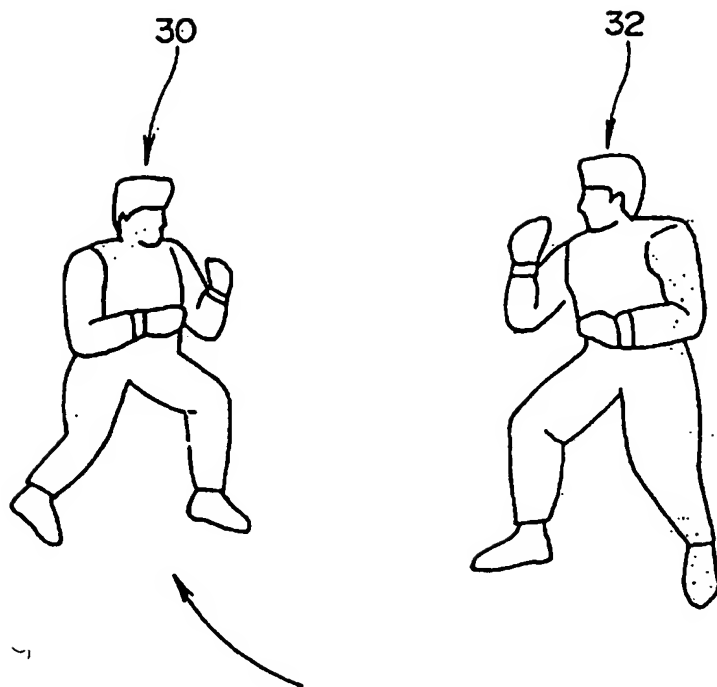
【図5】

5/60



[ 図 6 ]

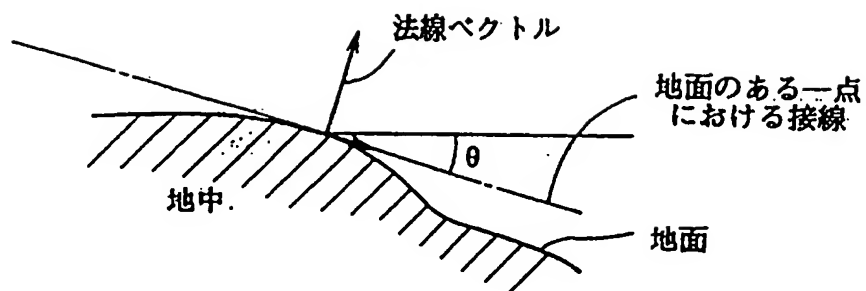
6/60



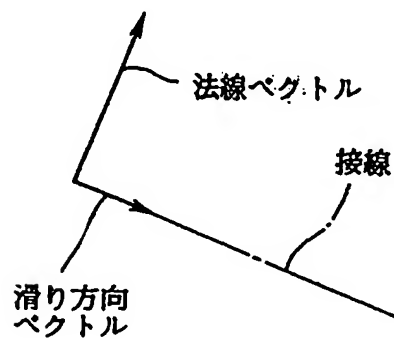
【図 7】

 $\frac{7}{60}$ 

(1)



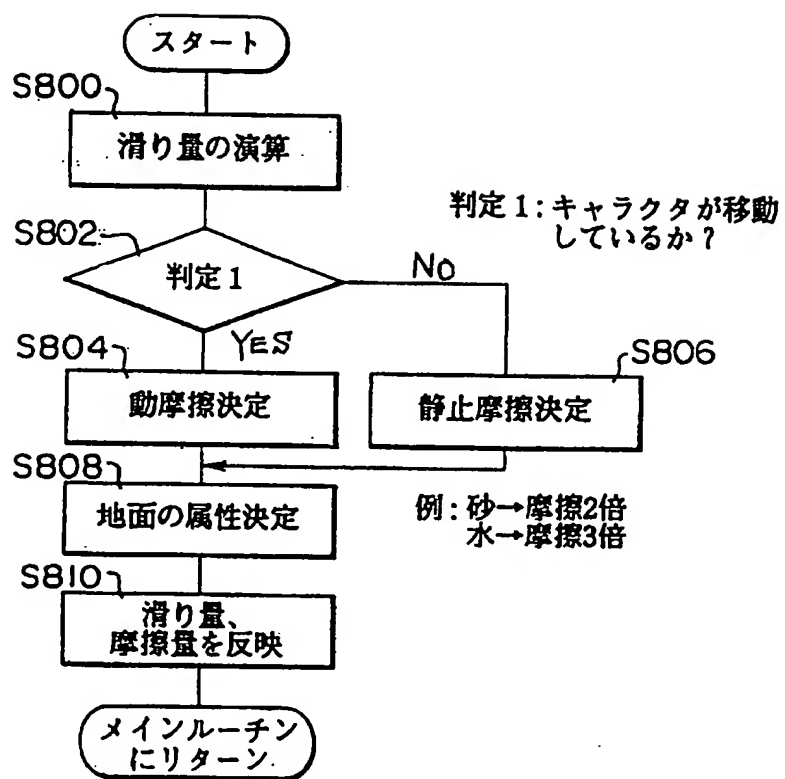
(2)





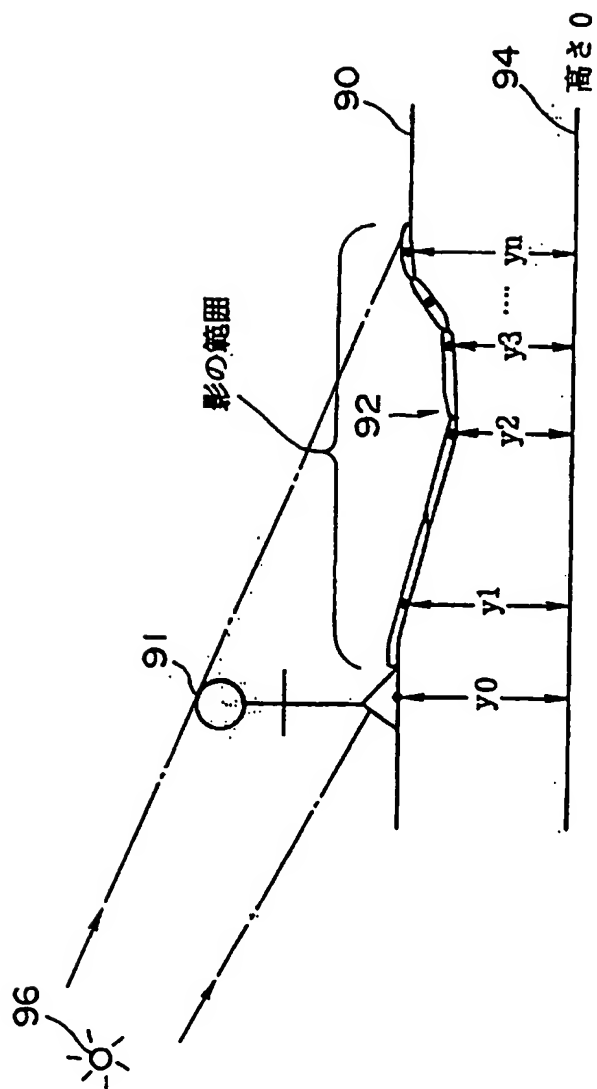
8/60

【図8】

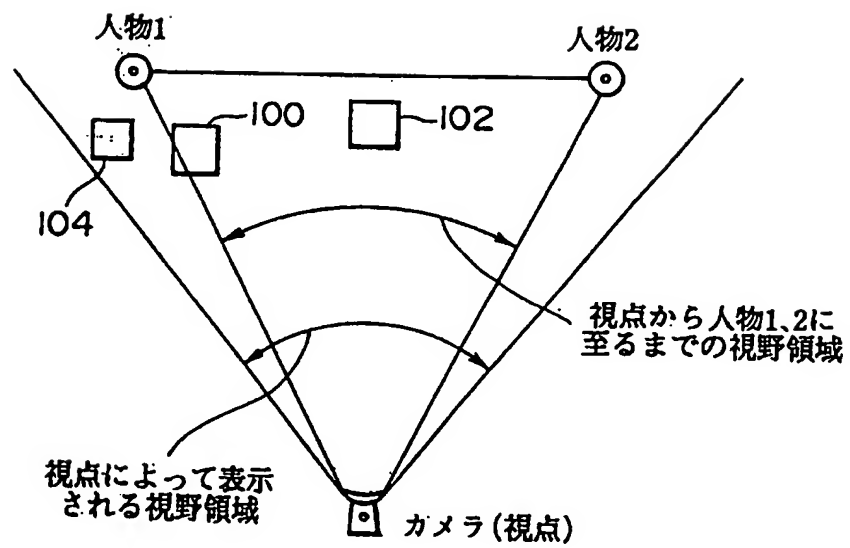


【図 9】

9/60

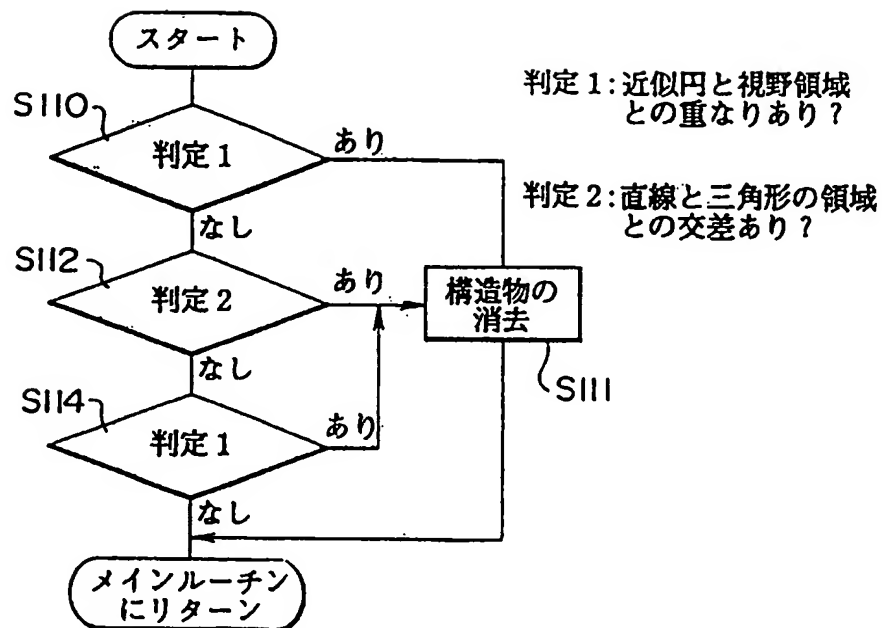


【図 10】

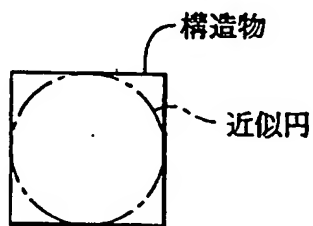
 $\frac{10}{60}$ 

11  
60

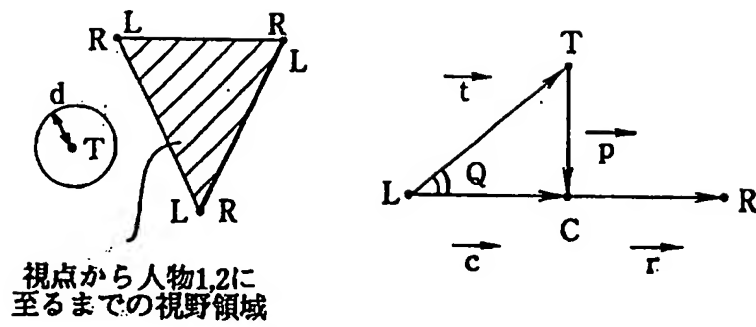
【図 11】



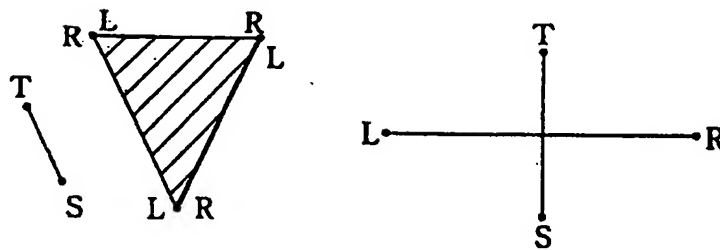
【図 12】



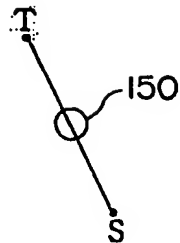
【図 13】



【図 14】

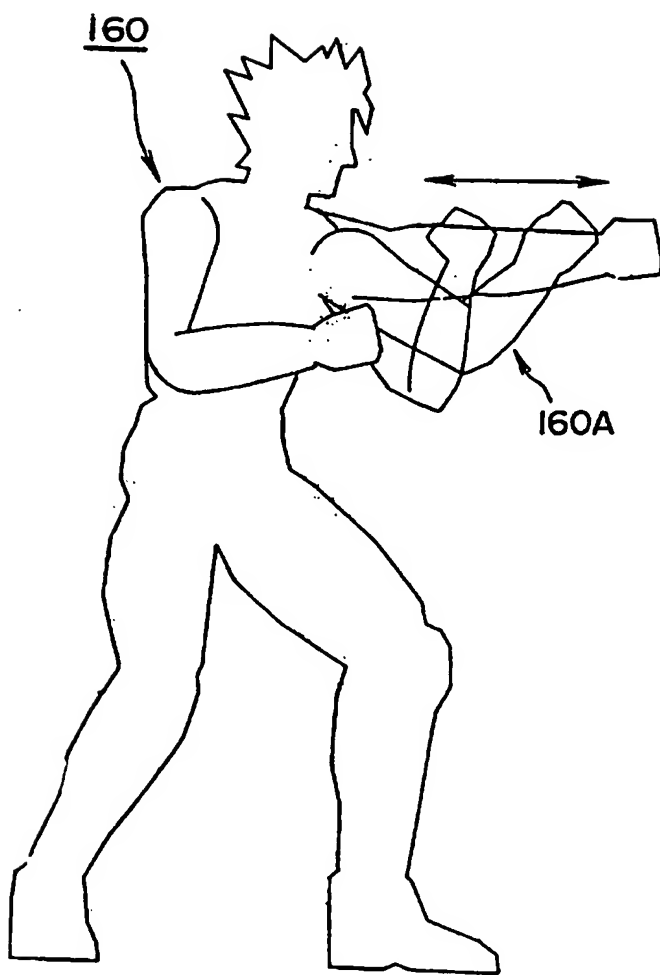


【図 15】



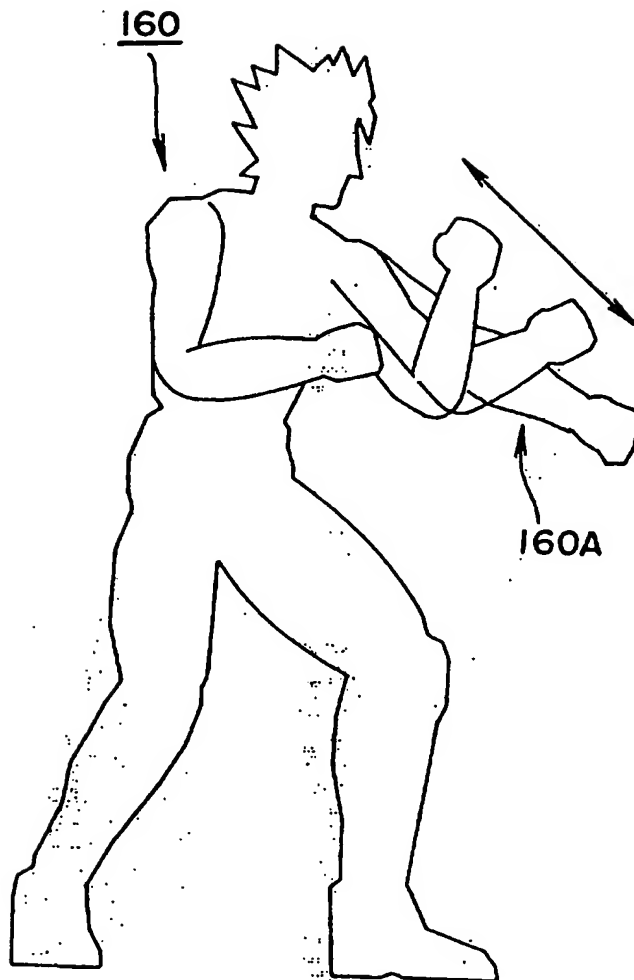
【 図 16 】

13/  
60

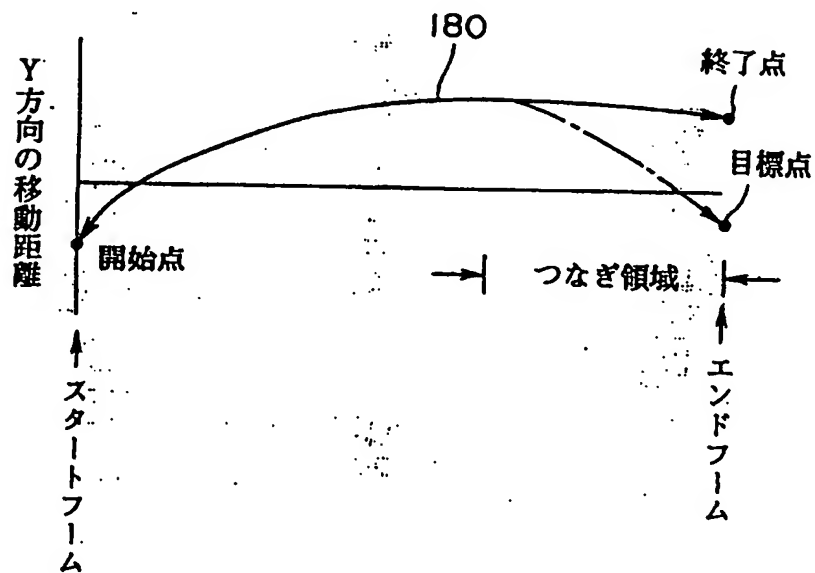
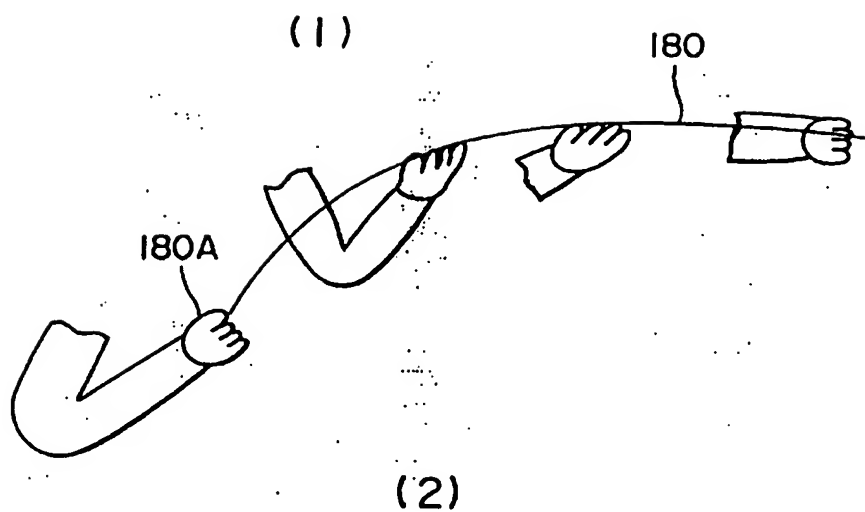


【 17 】

14  
/60

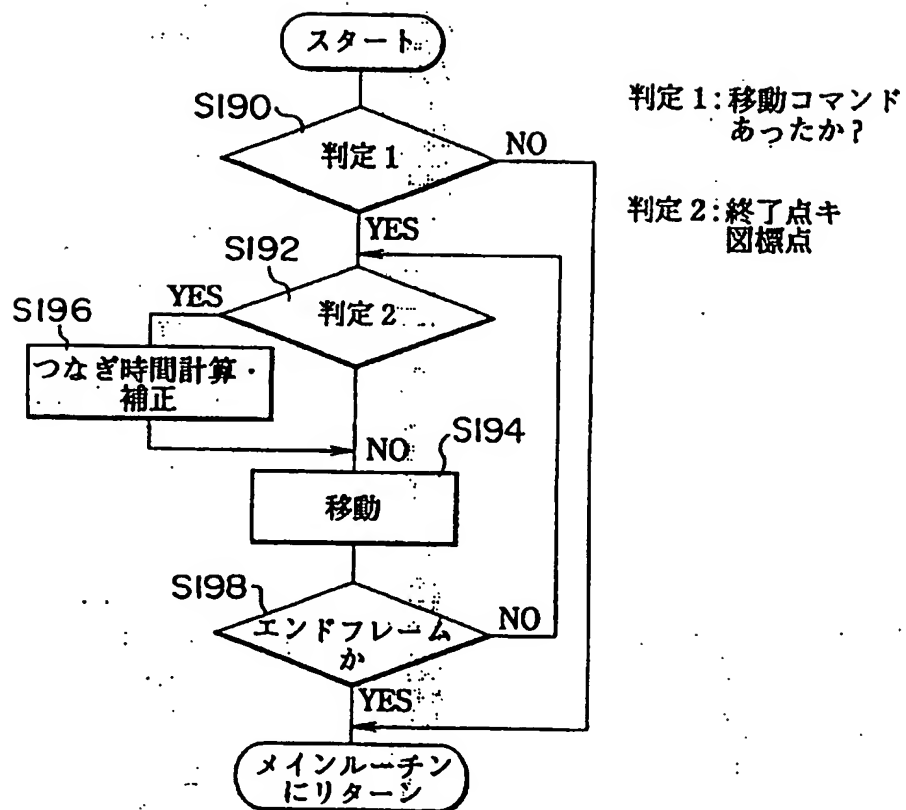


【図 18】

15  
/60

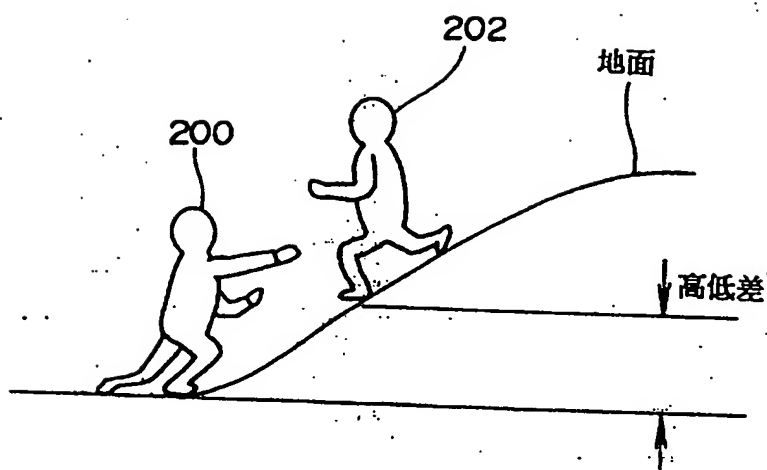


【図 19】

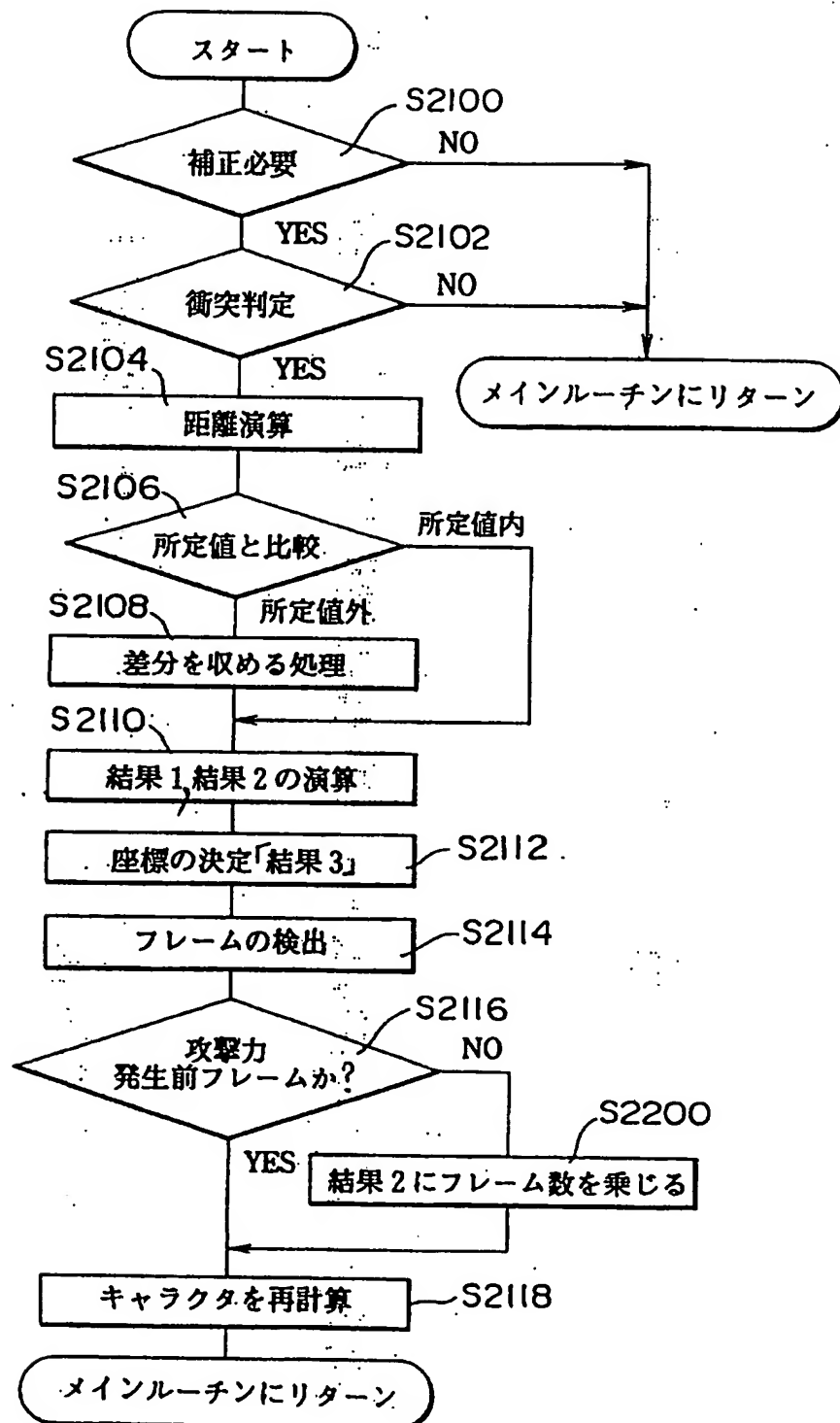
16/  
60

【 図 20 】

17  
/60

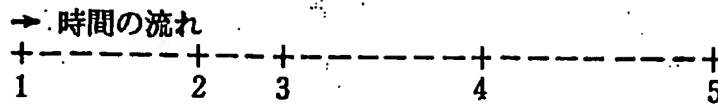


【図 21】

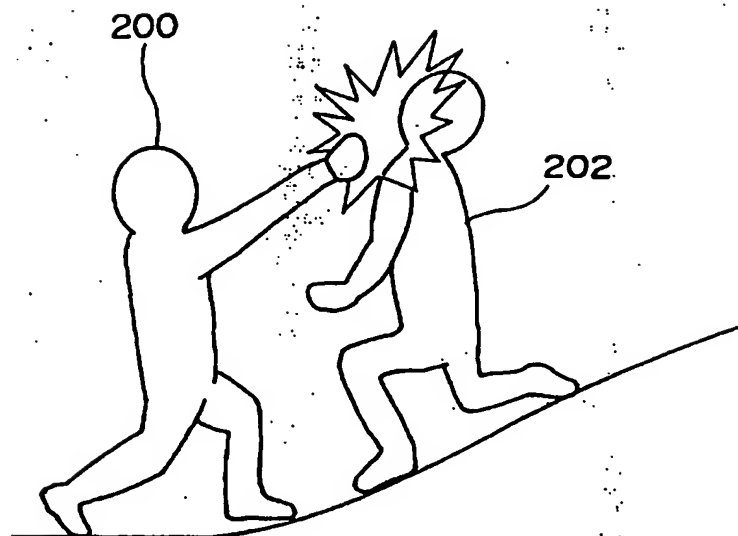
18  
/ 60

$\frac{19}{60}$ 

【図22】



【図23】

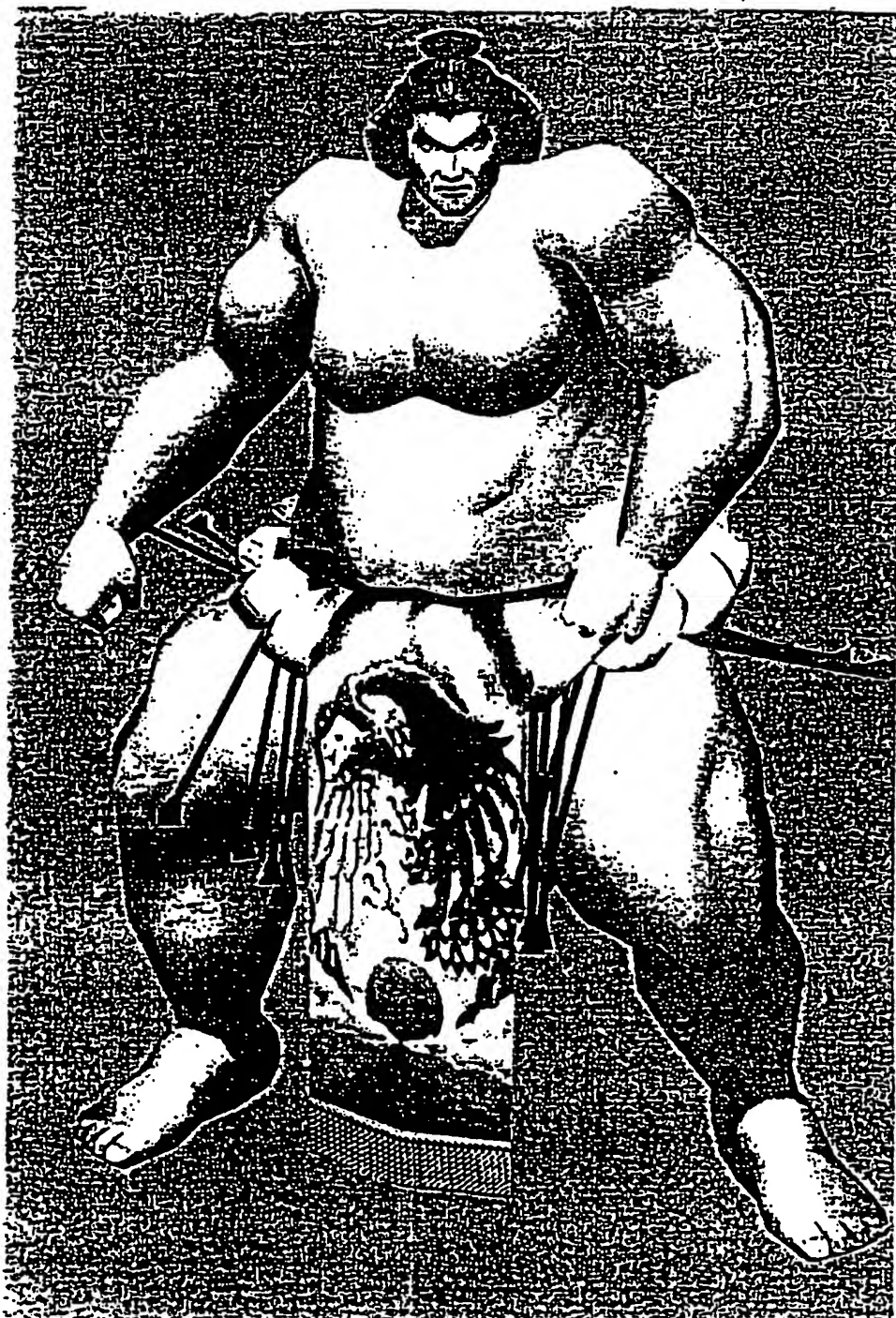


【図 24】



21/60

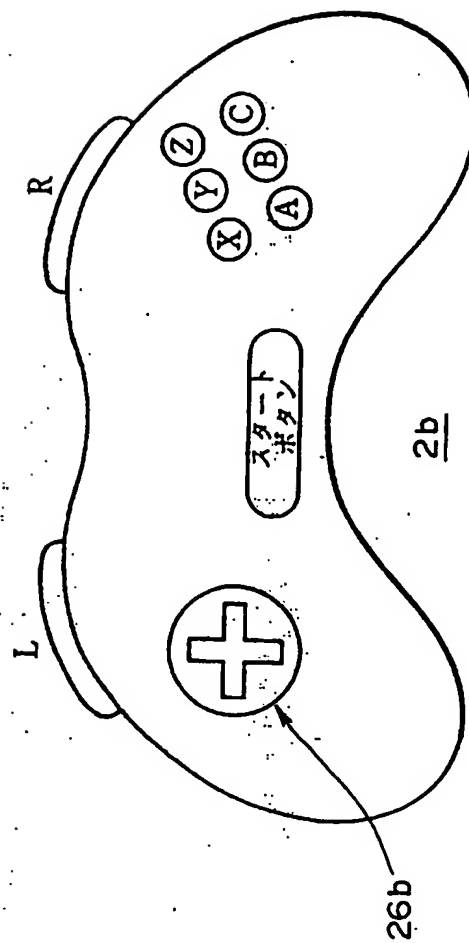
【図 25】



差替え用紙（規則26）

【 図 26 】

22  
60



23  
60

【図 27】

混合技		コマンド技	フレーム数
	1	ⓅⓅⓅ	002
	2	ⓅⓀⓖ	004
	3	Ⓟ + Ⓚ	020
		⋮	⋮
	n		
混合技ファイル # 1			登録ボタン # 1

Ⓟ (パンチ) : B ボタン

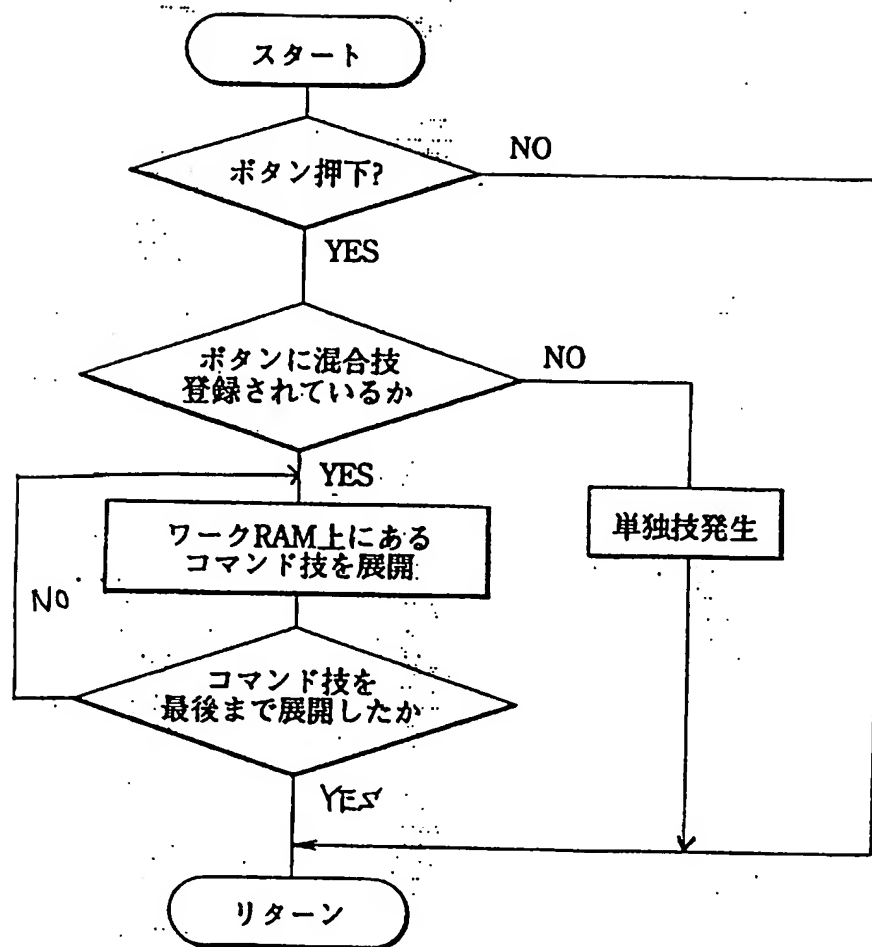
Ⓚ (キック) : C ボタン

ⓖ (ガード) : A ボタン

+ : 同時押し



【図 28】

24  
60



26 / 60

【図30】

# KEY ASSIGN

1P

EDIT

Alt-F G	使用しない
Alt-F P	スペシャル1
Alt-F R	スペシャル2
P + G	スペシャル3
R + G	スペシャル4
P + R	スペシャル5
P + R + G	

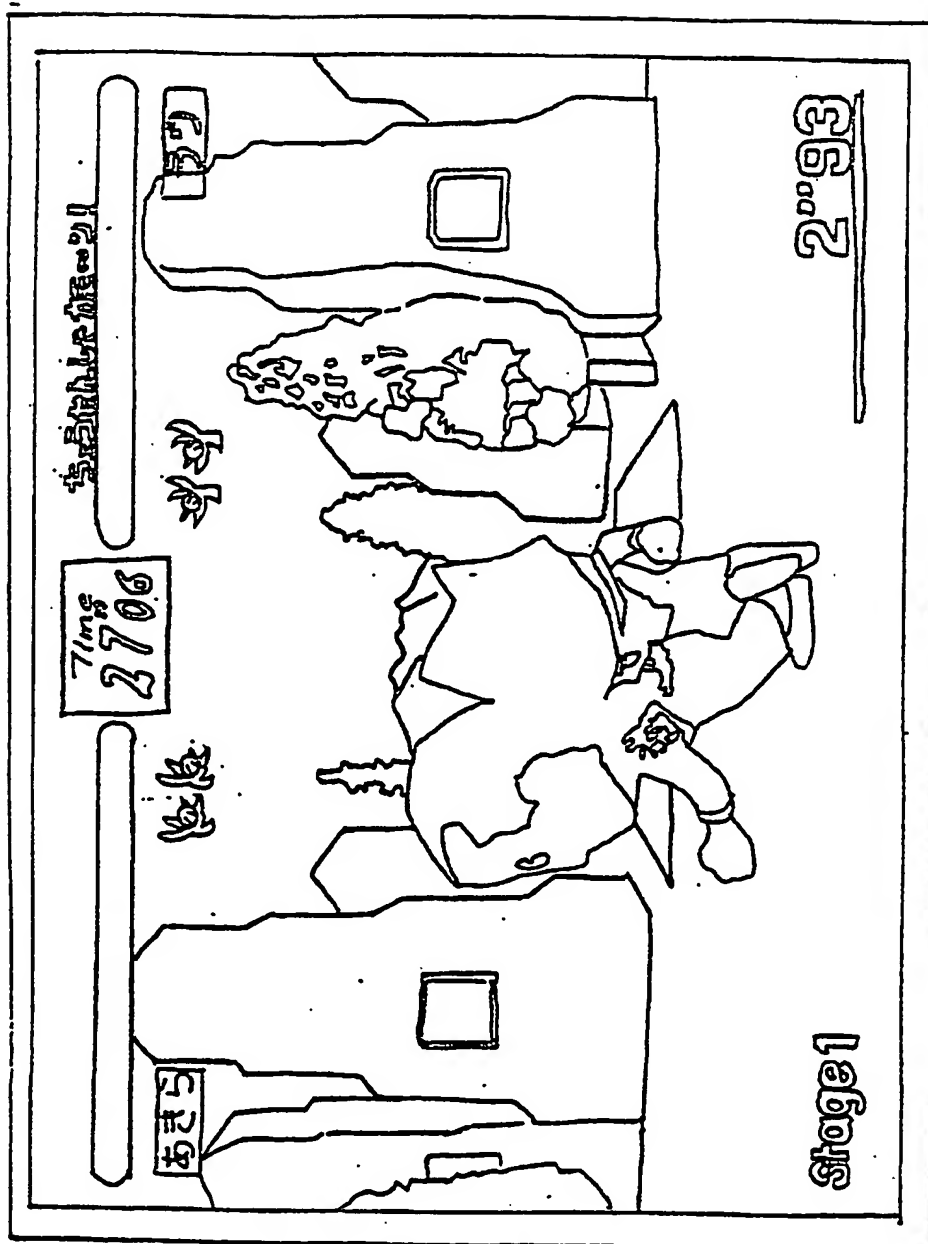
Alt-F G	
Alt-F P	
Alt-F R	
スペシャル1	
使用しない	
使用しない	
使用しない	
使用しない	
コンボ	

【図 3 1】

COMBO TUKURU		コンボ作り	
1	2	3	4
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20

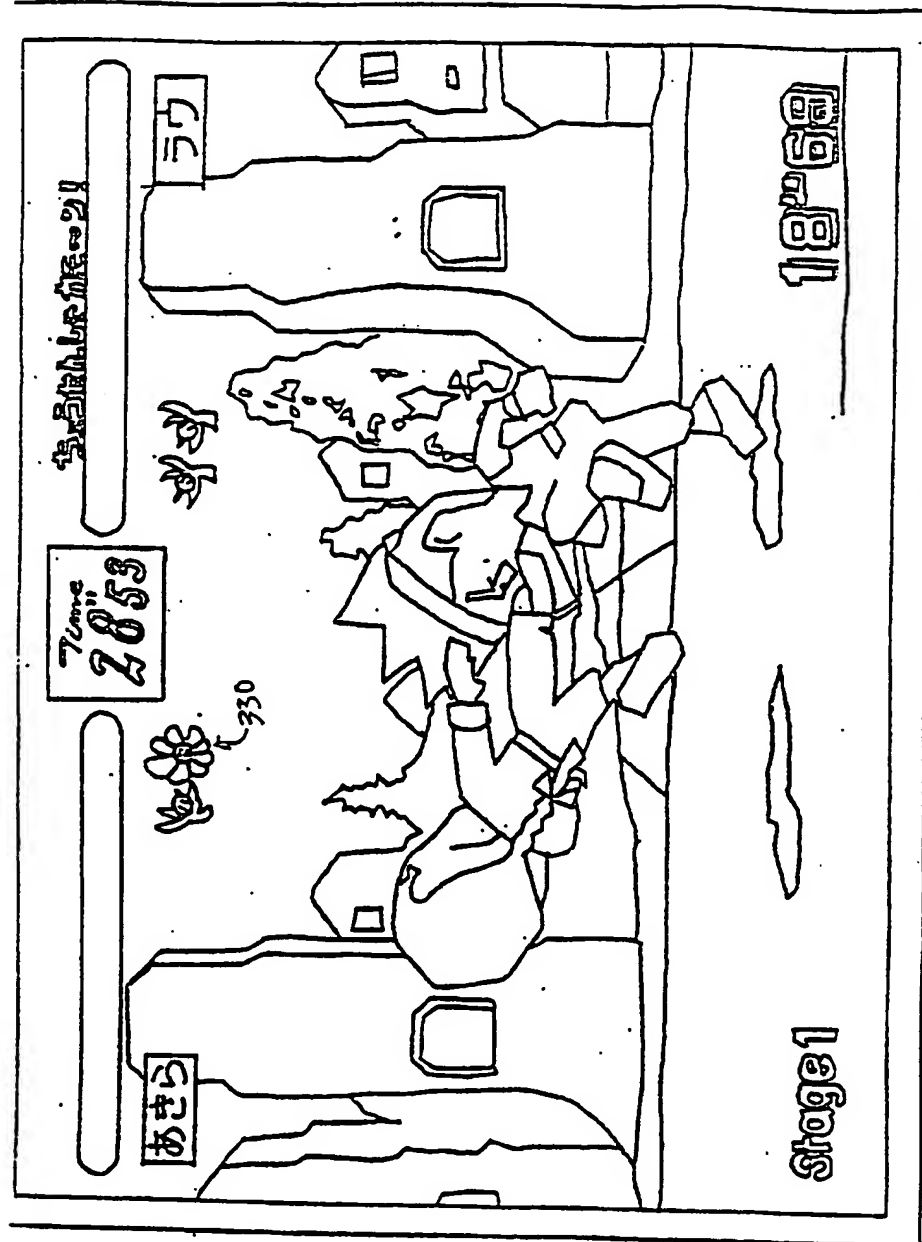
28 / 60

【図 3 2】



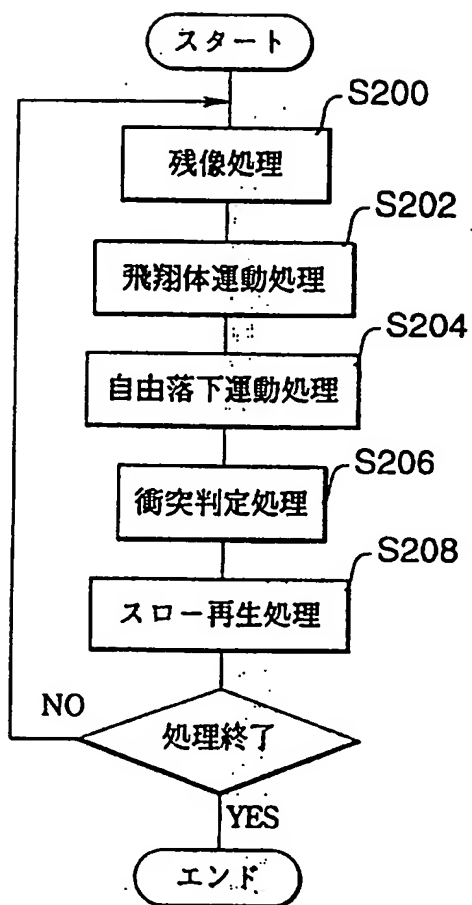
29/60

【図33】



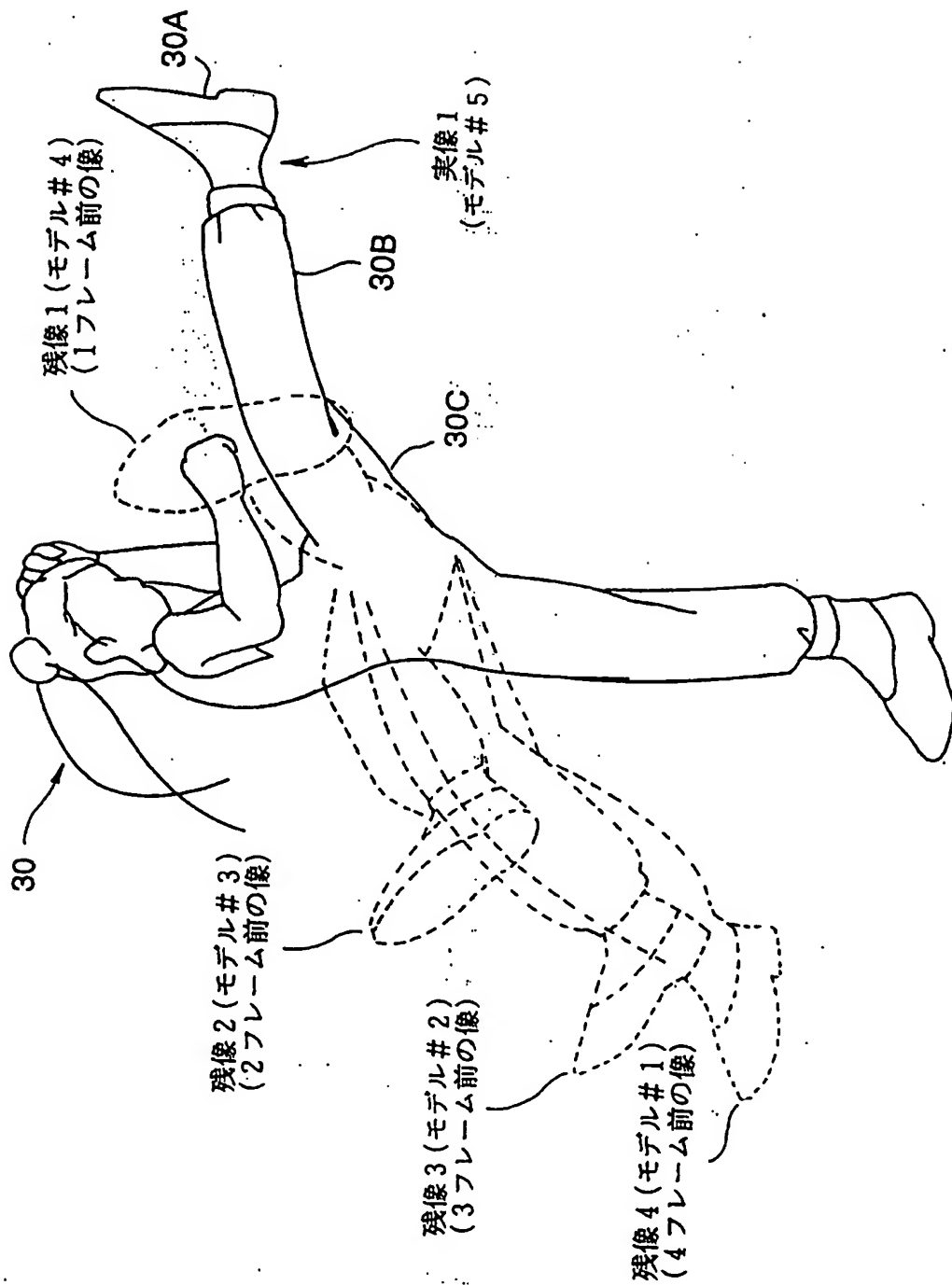
$\frac{30}{60}$ 

【図 34】



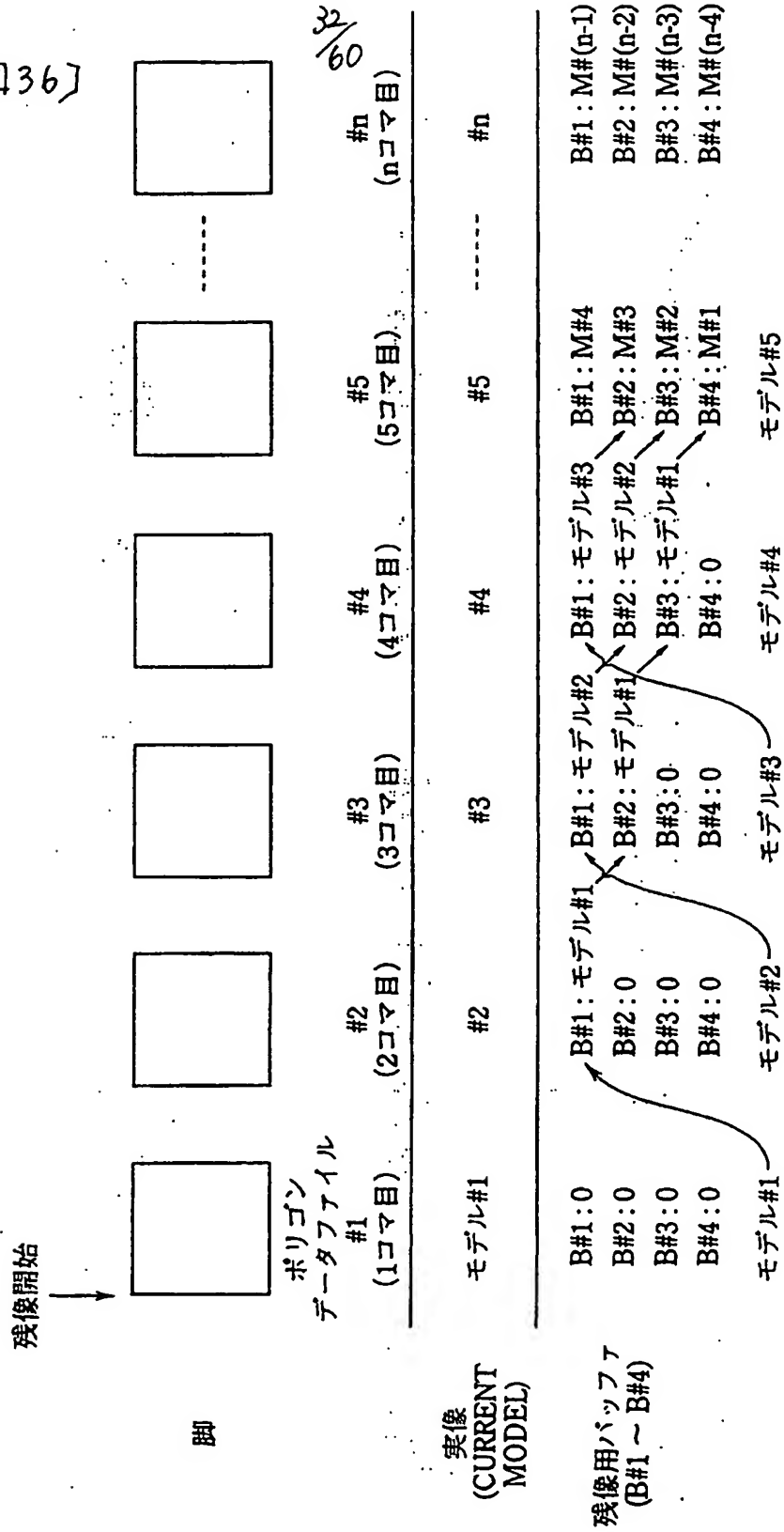
31/60

【図 35】



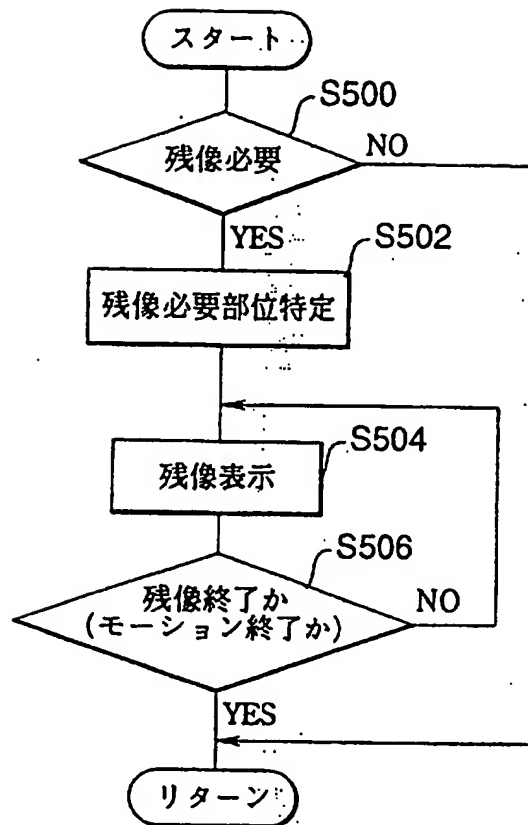


[図36]

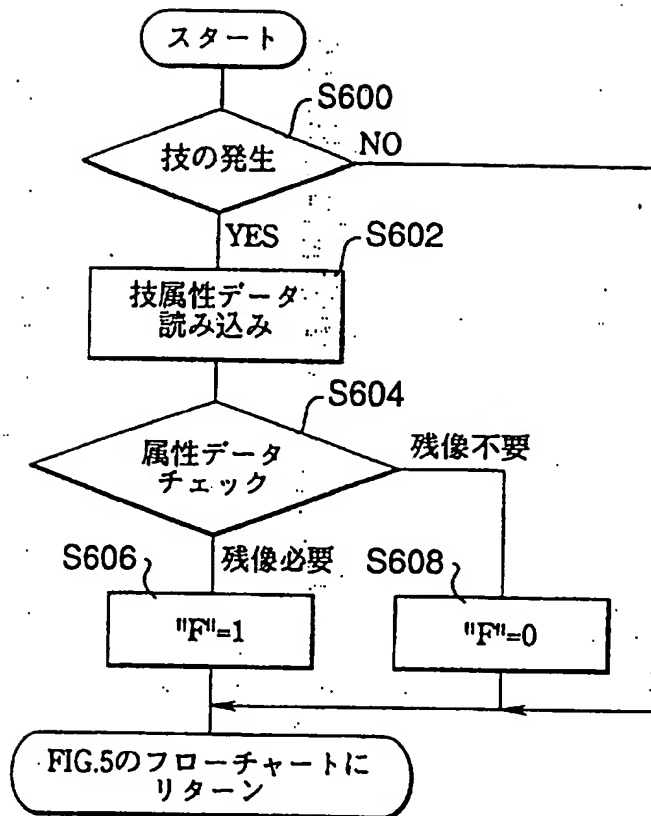


33  
60

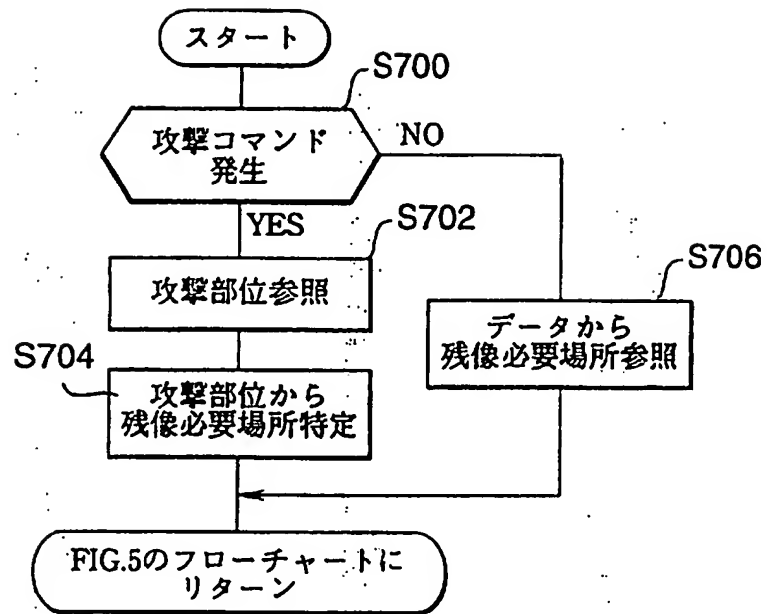
【図 37】



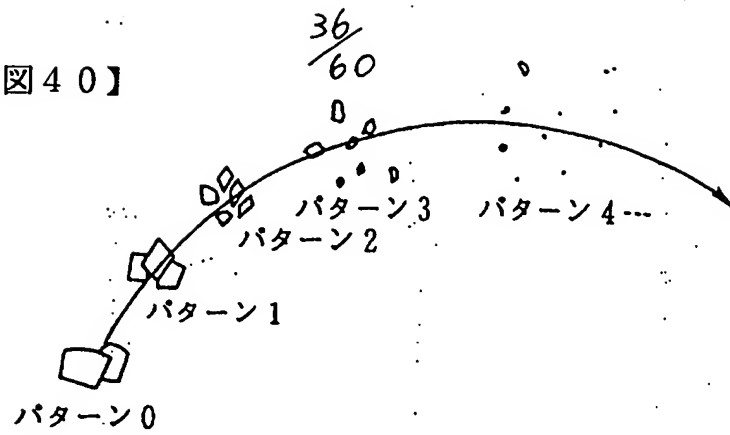
【図38】

34  
/60

【図 39】

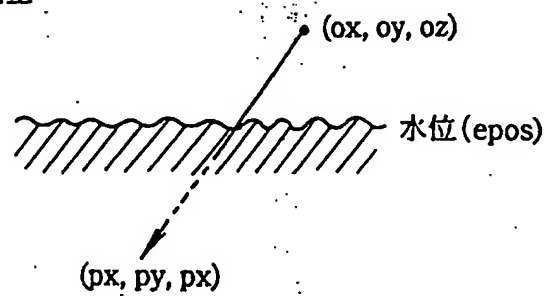
35  
60

【図 4 0】

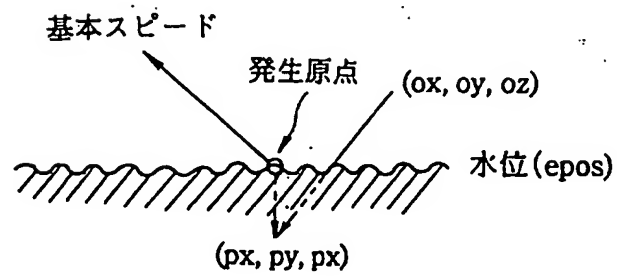


【図 4 1】

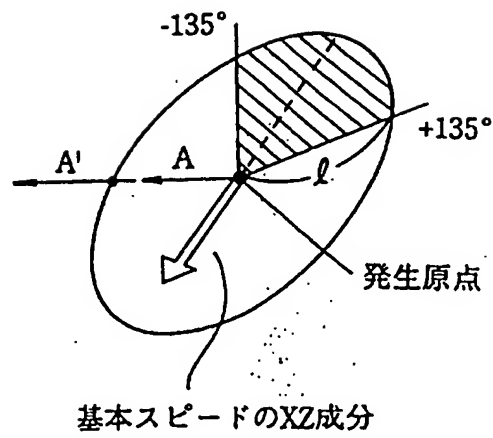
着地発生



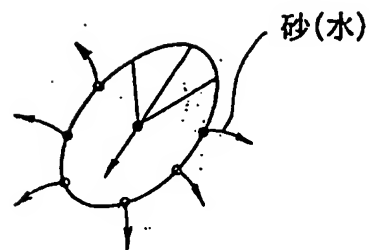
【図 4 2】



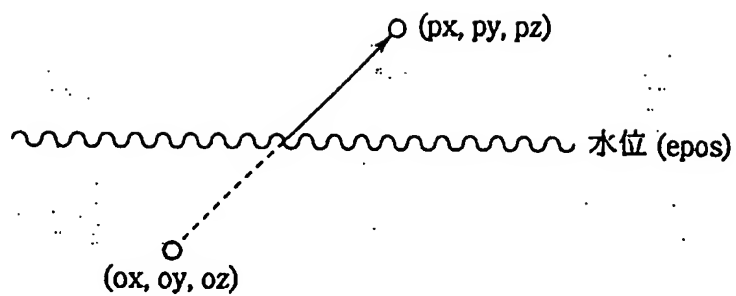
【図 4 3】

 $\frac{37}{60}$ 

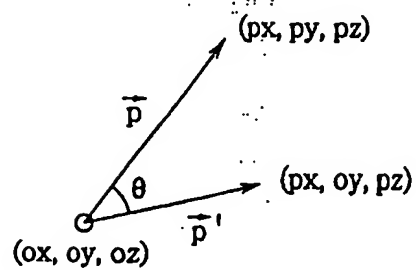
【図 4 4】



【図 4 5】

38  
60

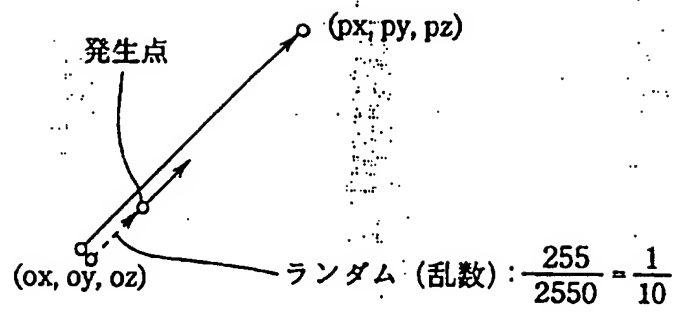
【図 4 6】



$$\cos\theta = \frac{\vec{p} \cdot \vec{p}'}{|\vec{p}| \cdot |\vec{p}'|}$$

39  
60

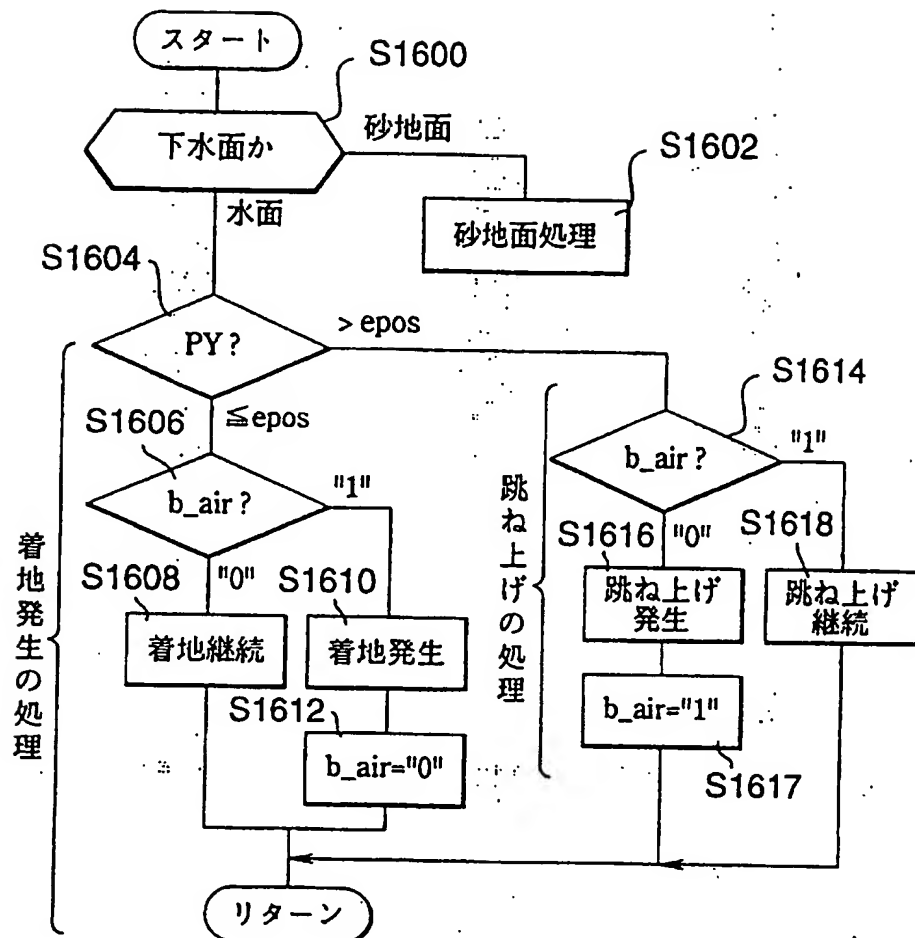
【図 47】



$$\text{発生点} : (ox + spd_x \times \frac{1}{10}, oy + spd_y \times \frac{1}{10}, oz + spd_z \times \frac{1}{10})$$



【図48】

40  
60

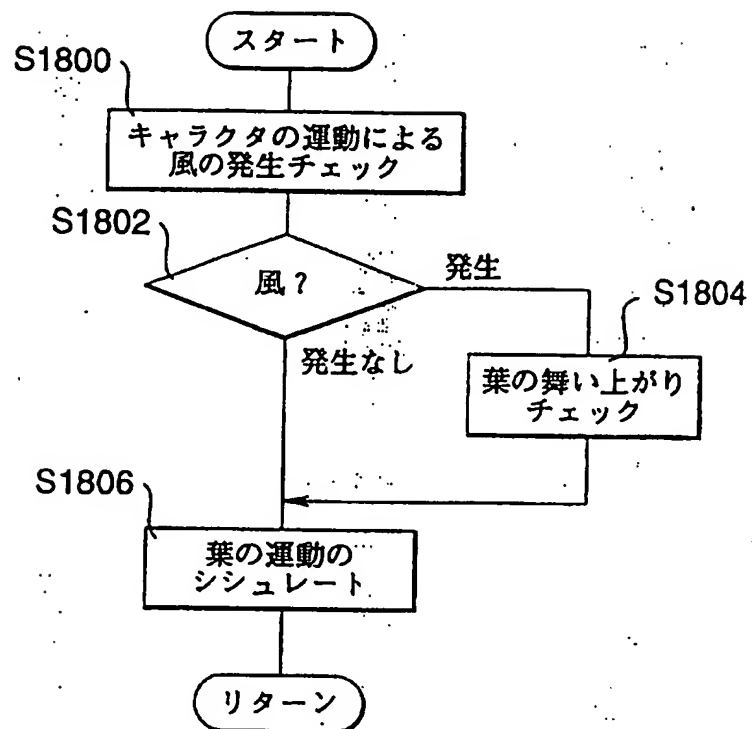
41  
60

【図 49】

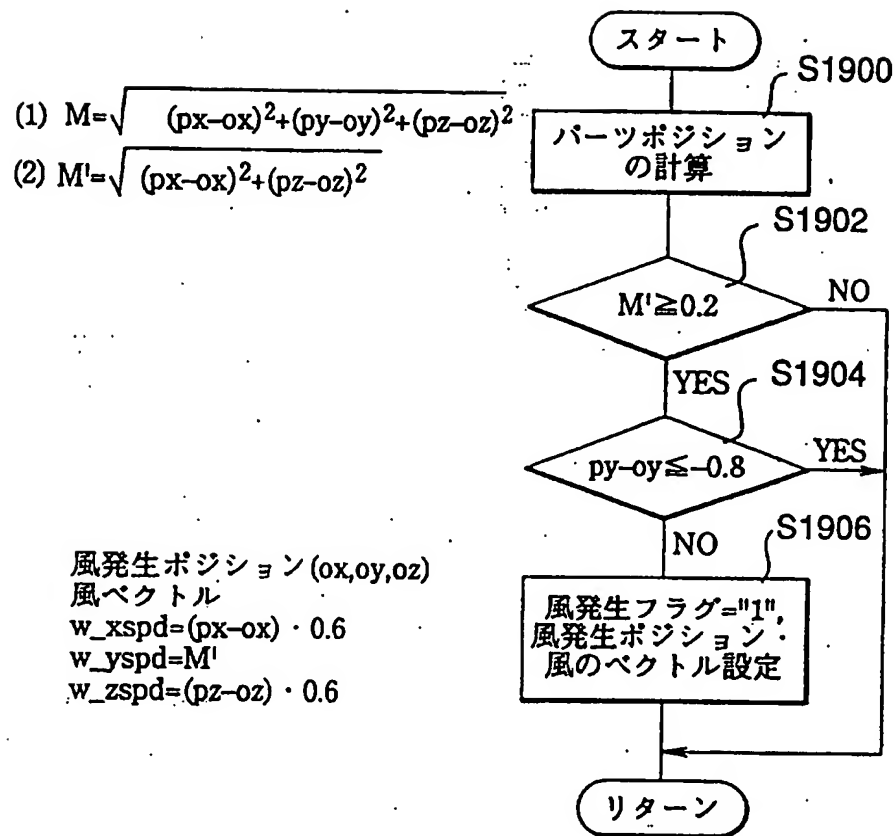


42  
/60

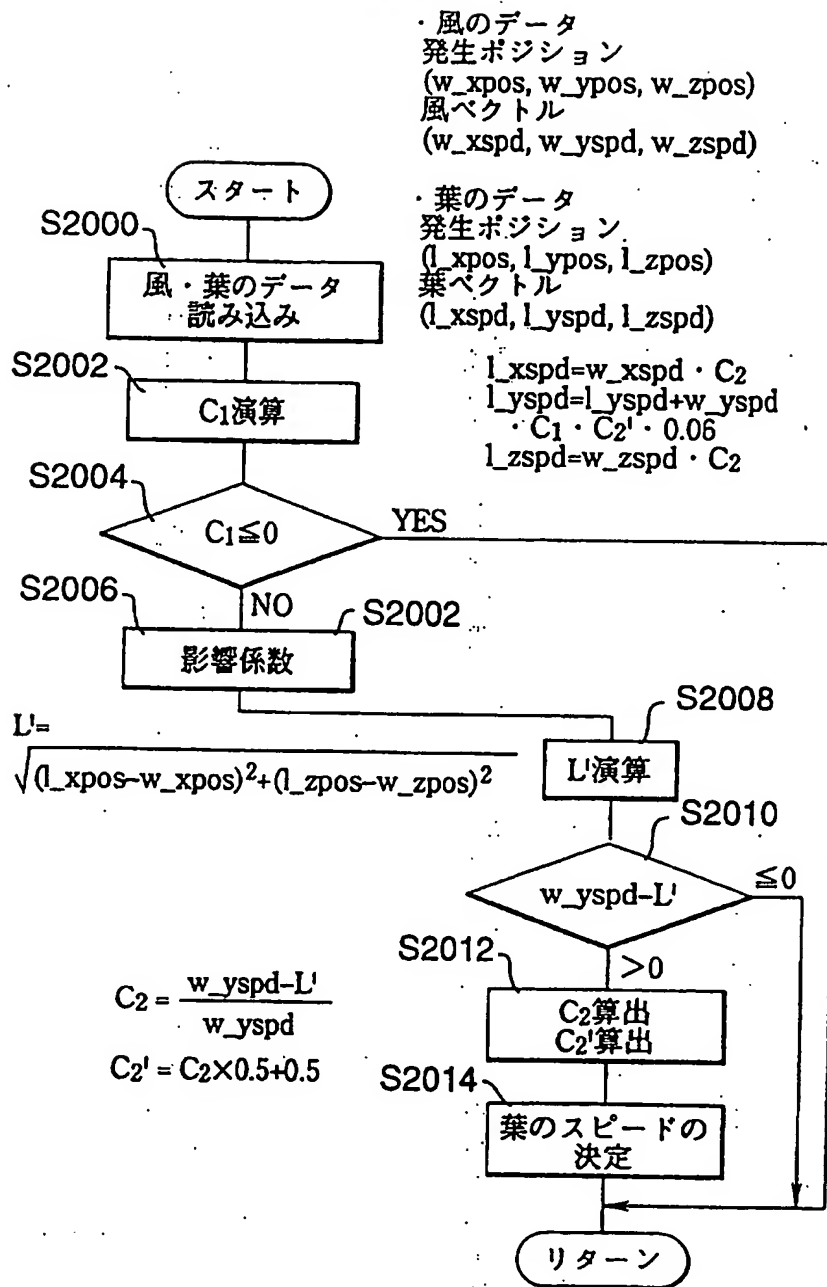
【図50】



【図 5 1】

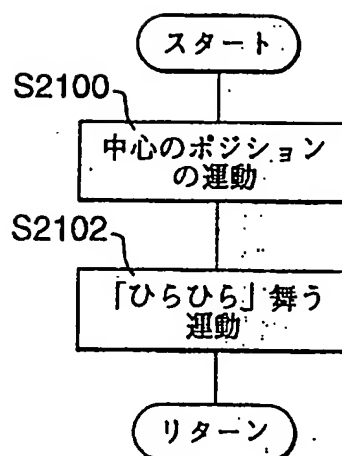
43  
60

【図 5 2】

44  
60

45  
/60

【図 5 3】



S2100

空中の場合

 $l\_xspd = l\_xspd \times 0.87$  $l\_yspd = l\_yspd - 0.02$  $l\_zspd = l\_zspd \times 0.87$ 

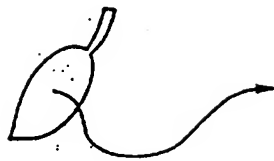
着地している場合

 $l\_xspd = l\_xspd \times 0.50$  $l\_yspd = 0$  $l\_zspd = l\_yspd \times 0.50$

【図54】

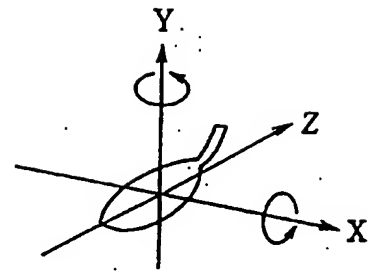
46  
/60

(1)



風の影響や重力によって葉の  
中心ポジションが移動する運動

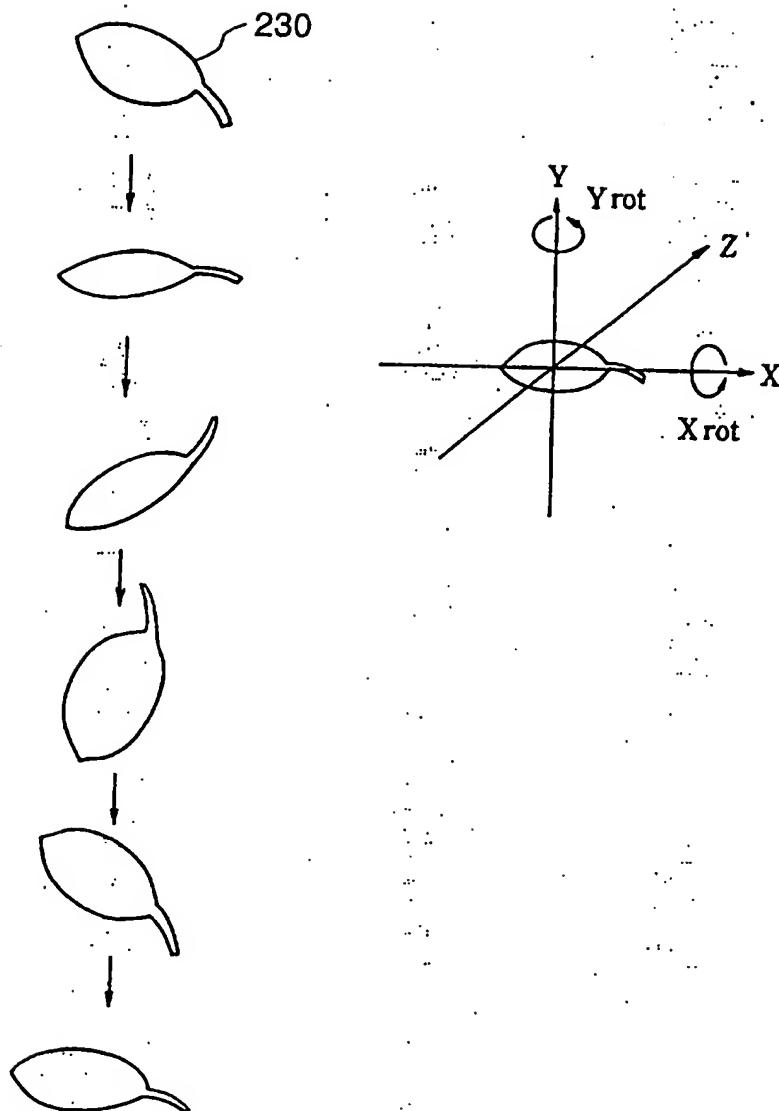
(2)



葉がひらひら舞う  
運動

【図 55】

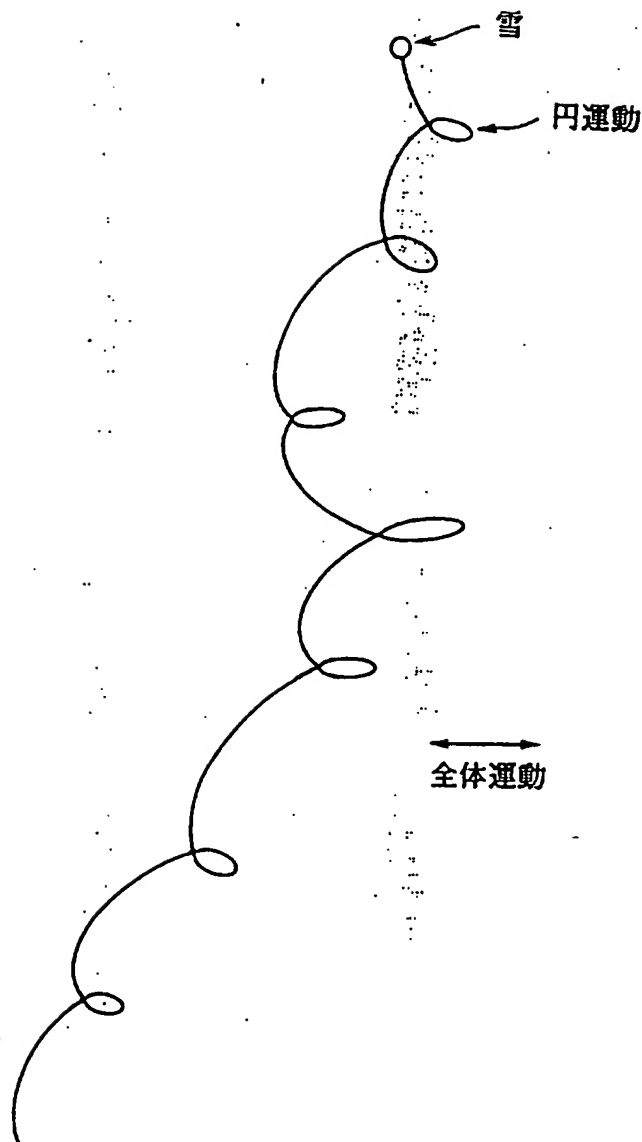
47  
60

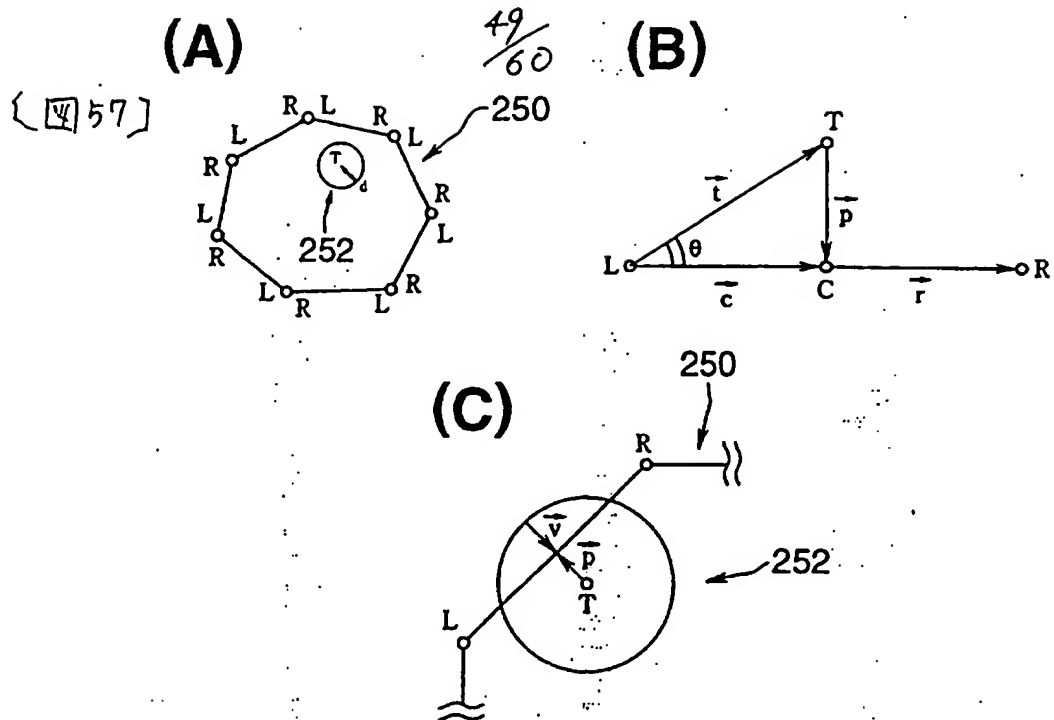




48  
/ 60

【図 5 6】





$$\vec{r} = \vec{R} - \vec{L}$$

$$\vec{t} = \vec{T} - \vec{L}$$

$$\vec{p} = \vec{C} + \vec{L} - \vec{T}$$

$$cx = rx \cdot \frac{|\vec{c}|}{|\vec{r}|}, \quad cz = rz \cdot \frac{|\vec{c}|}{|\vec{r}|}$$

ここで

$$|\vec{c}| = |\vec{t}| \cos \theta$$

内積の定理より

$$|\vec{r}| \cdot |\vec{t}| \cos \theta = rx tx + rz tz$$

$$\frac{|\vec{c}|}{|\vec{r}|} = \frac{rx tx + rz tz}{|\vec{r}|^2} = \frac{rx tx + rz tz}{rx^2 + rz^2}$$

ただし

$$0 < \frac{|\vec{c}|}{|\vec{r}|} < 1 \quad \text{--- (3)}$$

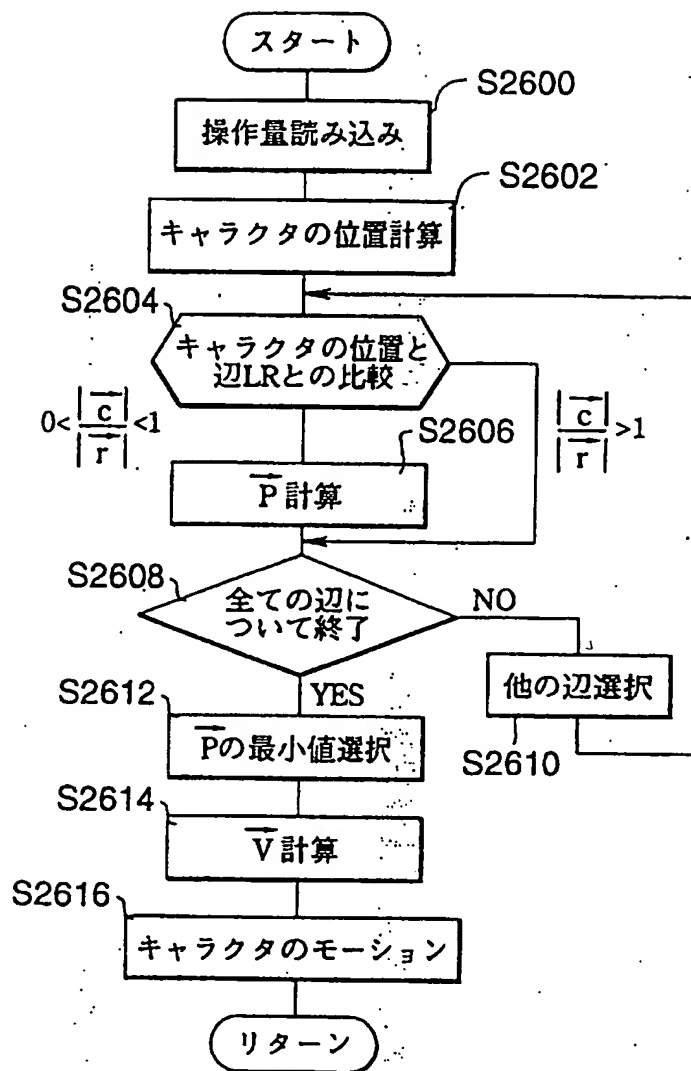
追い出しベクトルを  $\vec{v}$ ,  $\vec{p}$  の単位ベクトルを  $\vec{pe}$  とすると、  
点 T が直線 LR の正領域にある場合

$$\vec{v} = (|\vec{p}| + d) \cdot \vec{pe} \quad \text{--- (1)}$$

点 T が直線 LR の負領域にある場合

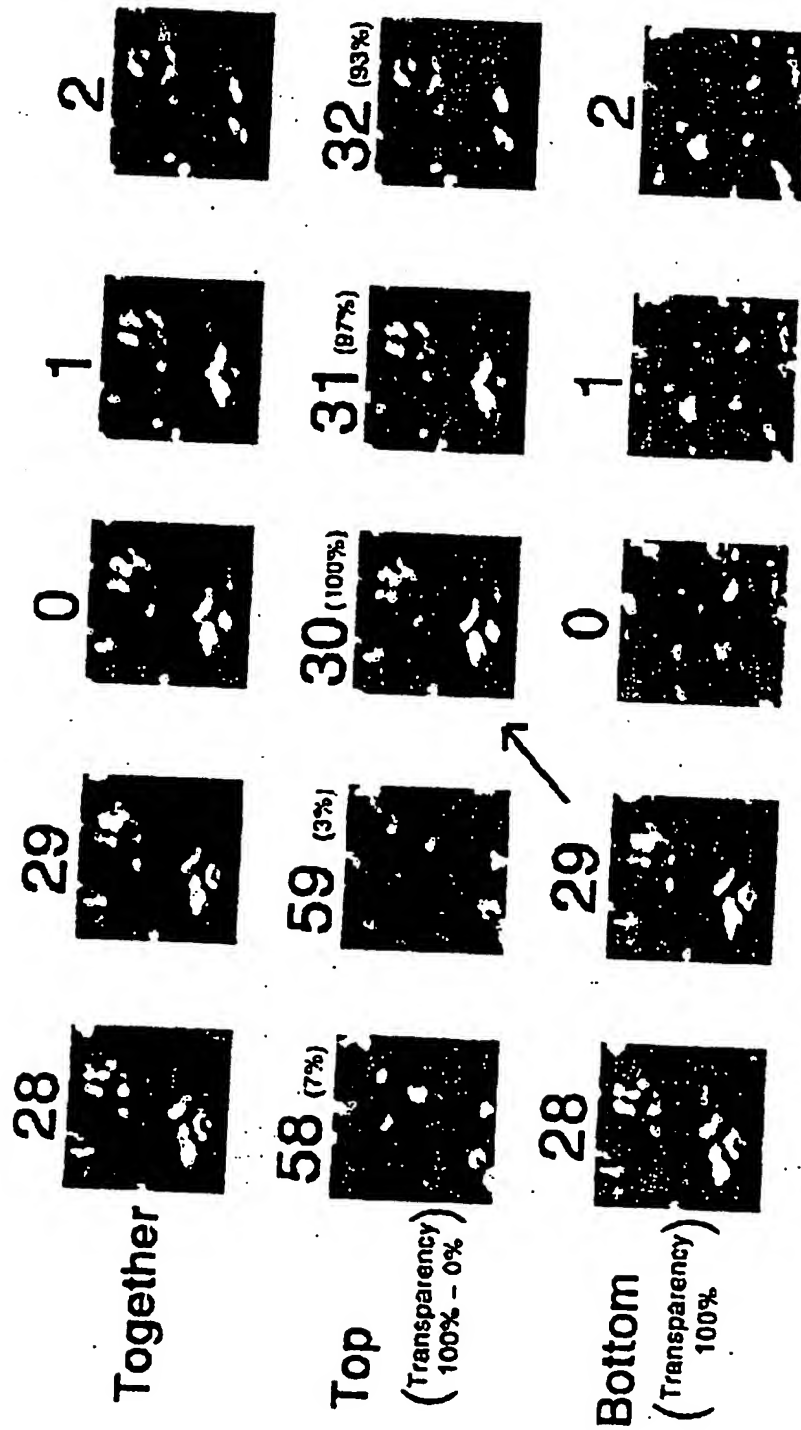
$$\left. \begin{array}{l} |\vec{p}| - d < 0 \text{ ならば } \vec{v} = (|\vec{p}| - d) \cdot (-\vec{pe}) \\ |\vec{p}| - d \geq 0 \text{ ならば } \vec{v} = \vec{0} \end{array} \right\} \quad (2)$$

【図58】

50/  
60

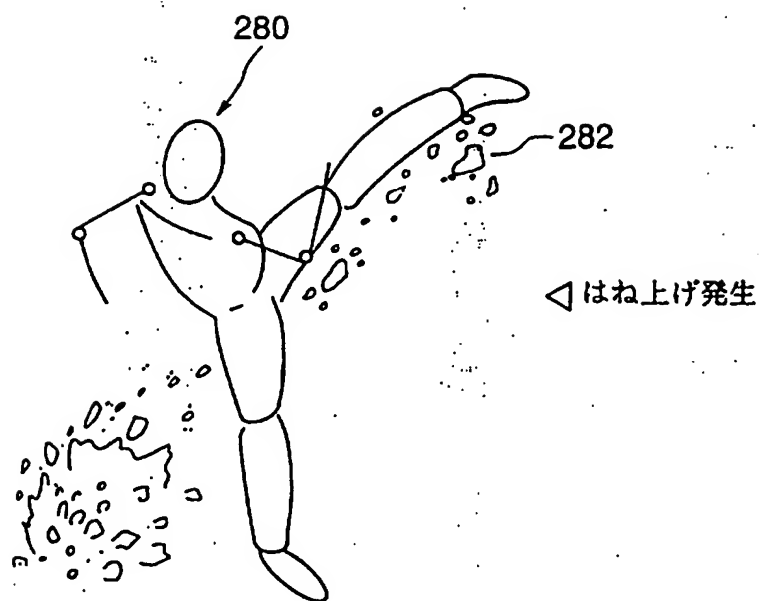
【図 59】

$\frac{51}{60}$

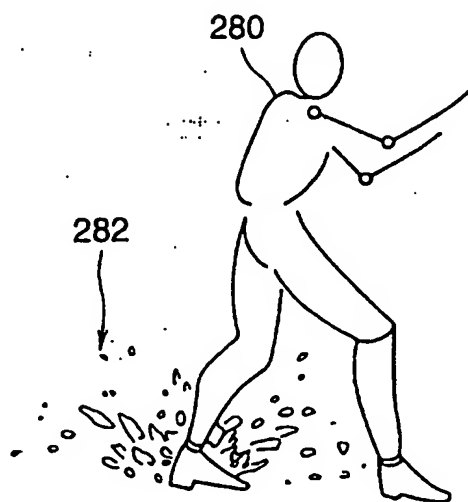


52  
60

【図 60】

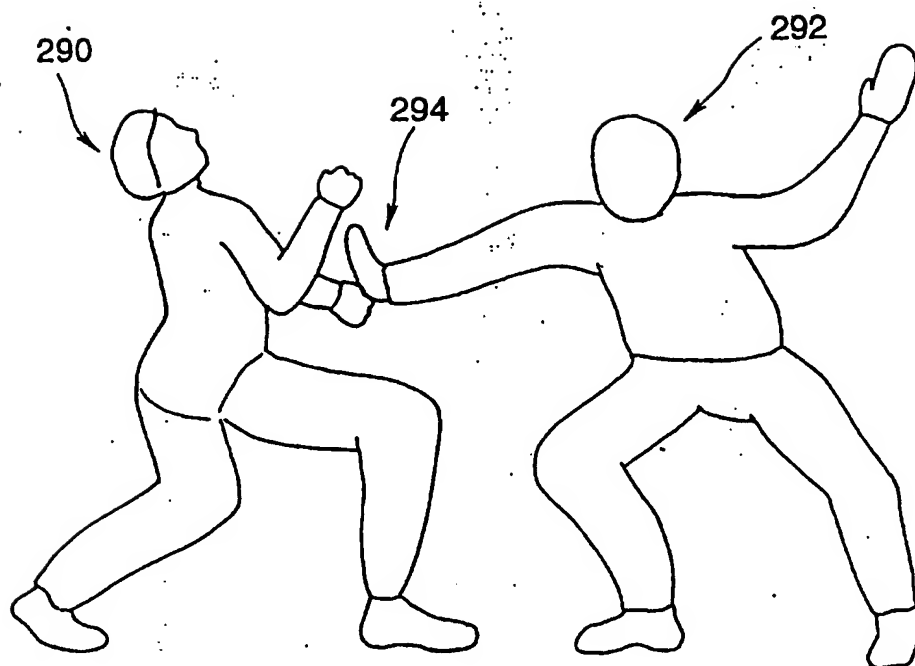


着地発生 ▷



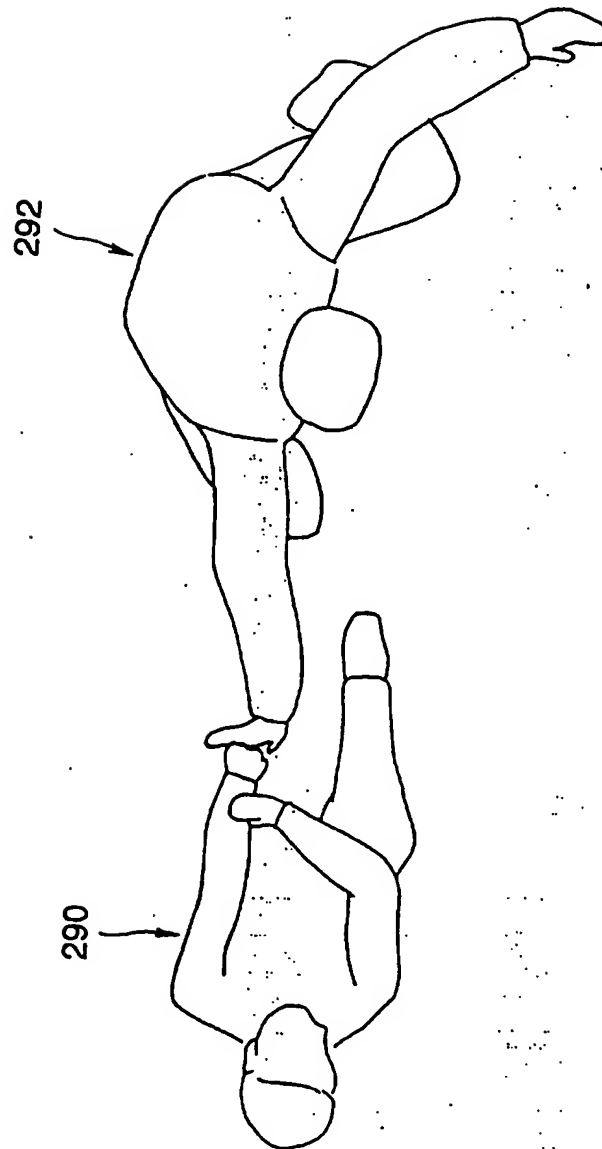
【図 6 1】

53  
60



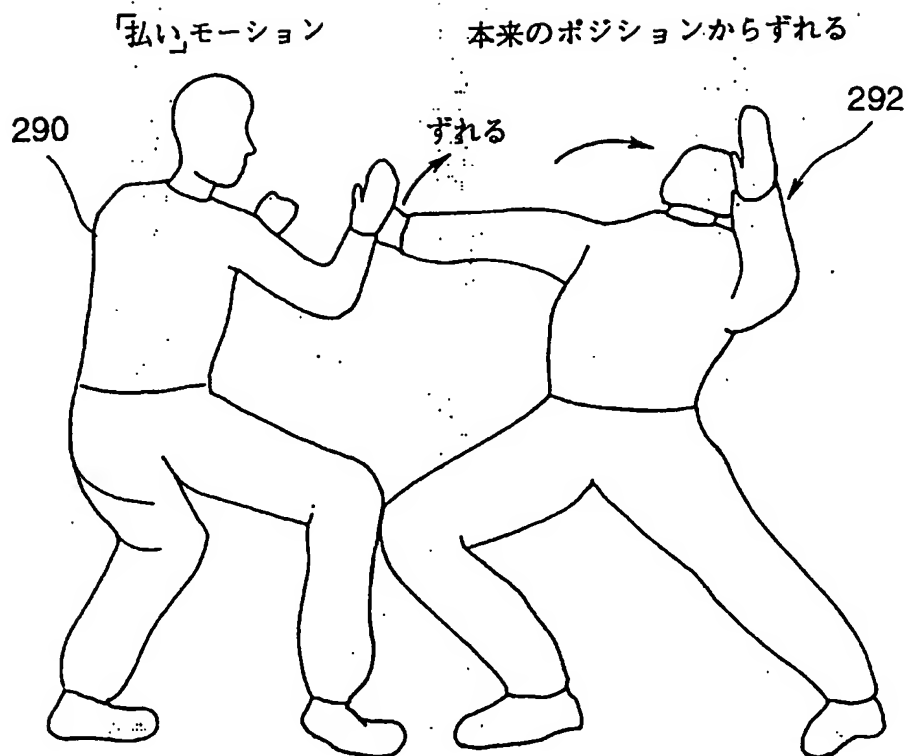
【図 6 2】

54/  
60



【図63】

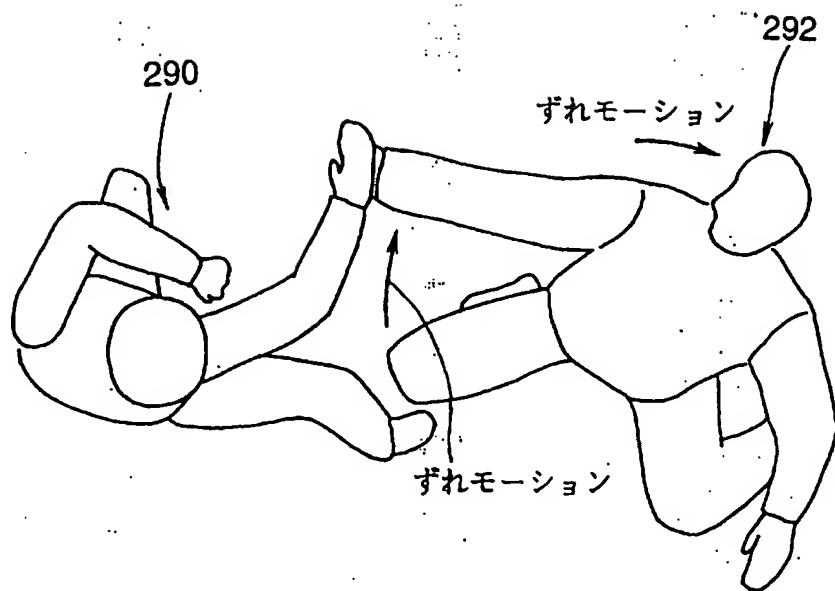
55  
/60





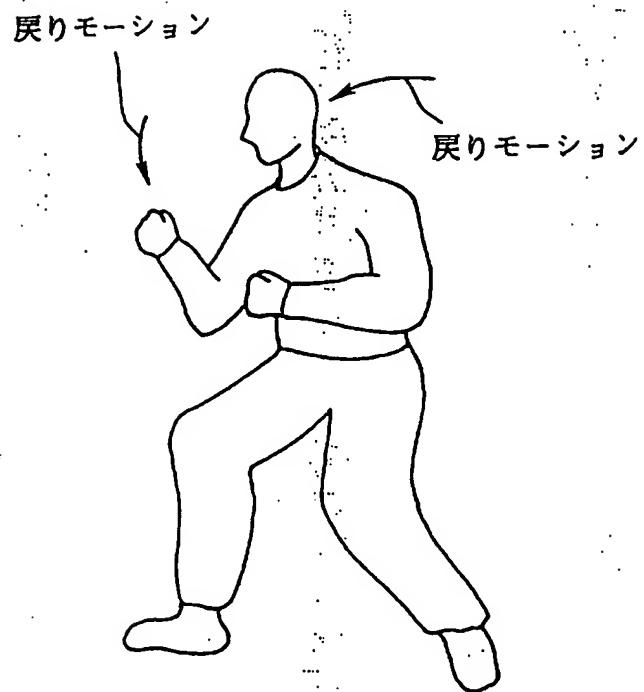
【図 64】

56/  
60

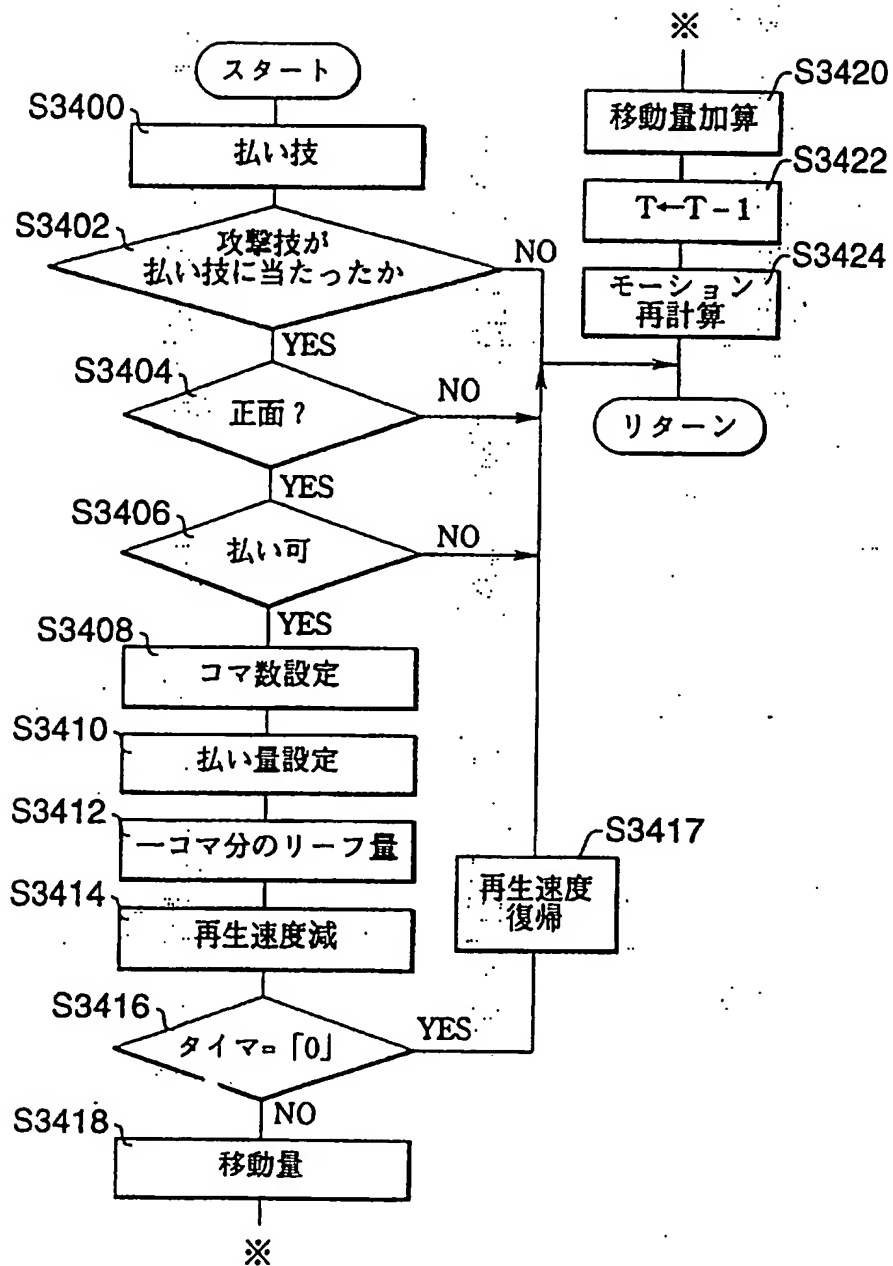


【図 65】

57  
/ 60

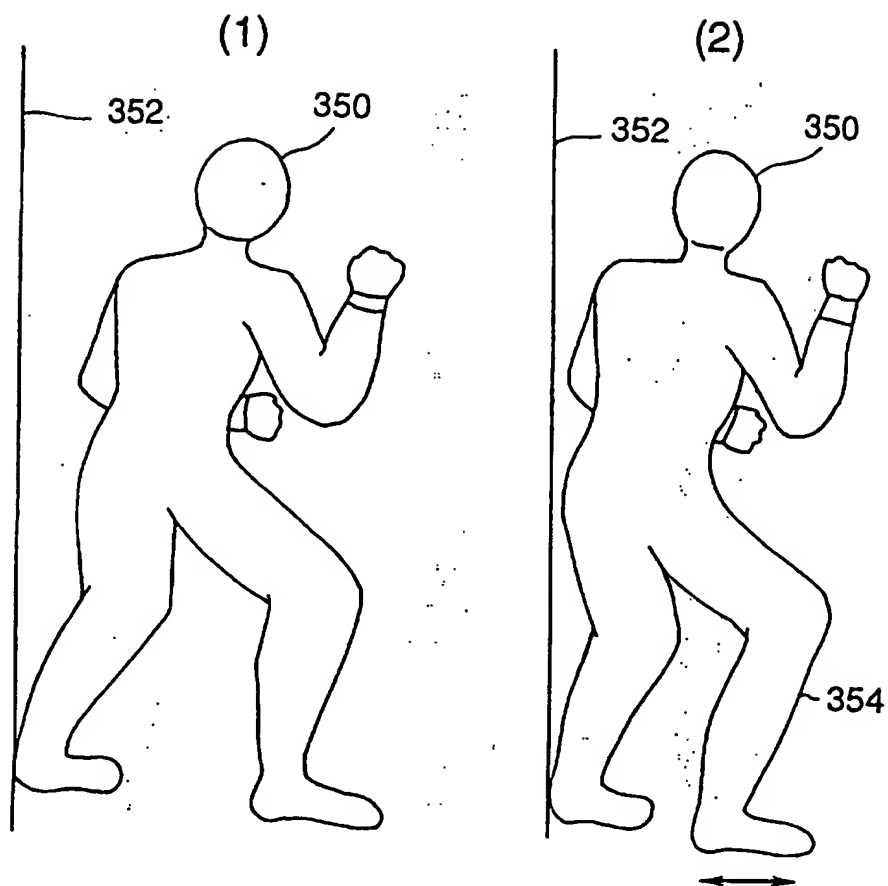


【図 66】

58  
60

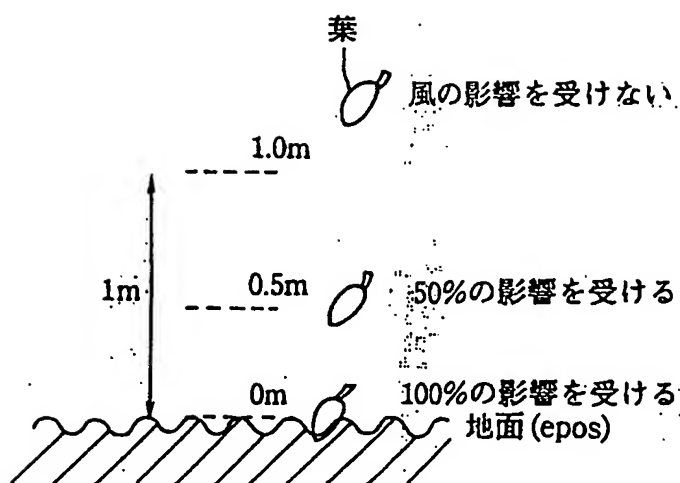
【図 67】

59/  
60



$\frac{60}{60}$ 

【図 68】



## INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP97/01896

## A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl<sup>6</sup> A63F9/22, G06T15/70

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int. Cl<sup>6</sup> A63F9/22, G06T15/70

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922 - 1996	Jitsuyo Shinan Toroku
Kokai Jitsuyo Shinan Koho	1971 - 1997	Koho
Toroku Jitsuyo Shinan Koho	1994 - 1997	1996 - 1997

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JP, 6-277363, A (Namco Ltd.), October 4, 1994 (04. 10. 94) (Family: none) Column 9, line 47 to column 11, line 47; Figs. 9, 10	1, 2, 17
Y	Full descriptions; Figs. 1 to 24	8 - 10
A	Full descriptions; Figs. 1 to 24	3-7, 11-16
A	JP, 7-116343, A (Sega Enterprises, Ltd.), May 9, 1995 (09. 05. 95) (Family: none) Full descriptions; Figs. 1 to 6	1 - 17
A	JP, 6-277362, A (Namco Ltd.), October 4, 1994 (04. 10. 94) (Family: none) Full descriptions; Figs. 1 to 29	7
Y	JP, 7-328228, A (Sega Enterprises, Ltd.), December 19, 1995 (19. 12. 95) (Family: none) Full descriptions; Figs. 1 to 6	8 - 10
Y	JP, 7-191662, A (Konami Co., Ltd.), July 28, 1995 (28. 07. 95) (Family: none)	8 - 10

☒ Further documents are listed in the continuation of Box C.☐ See patent family annex.

## \* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&amp;" document member of the same patent family

Date of the actual completion of the international search

September 2, 1997 (02. 09. 97)

Date of mailing of the international search report

September 9, 1997 (09. 09. 97)

Name and mailing address of the ISA/

Japanese Patent Office

Facsimile No.

Authorized officer

Telephone No.

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP97/01896

## C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	Full descriptions; Figs. 1 to 5	
A	JP, 8-137378, A (Hitachi Denshi, Ltd.), May 31, 1996 (31. 05. 96) (Family: none) Full descriptions; Fig. 1	11 - 12
A	JP, 7-105401, A (Fujitsu Ltd.), April 21, 1995 (21. 04. 95) (Family: none) Full descriptions; Figs. 1 to 18	14 - 15
P,A	JP, 8-202342, A (Capcom Co., Ltd.), August 9, 1996 (09. 08. 96) (Family: none) Full descriptions; Figs. 1 to 7	16

A. 発明の属する分野の分類 (国際特許分類 (IPC))		
Int. Cl <sup>6</sup> A63F9/22, G06T15/70		
B. 調査を行った分野		
調査を行った最小限資料 (国際特許分類 (IPC))		
Int. Cl <sup>6</sup> A63F9/22, G06T15/70		
最小限資料以外の資料で調査を行った分野に含まれるもの 日本国実用新案公報 1922-1996年 日本国公開実用新案公報 1971-1997年 日本国登録実用新案公報 1994-1997年 日本国実用新案登録公報 1996-1997年		
国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)		
C. 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
X Y A	JP, 6-277363, A (株式会社ナムコ) 4. 10月. 1994 (04. 10. 94) (ファミリーなし) 第9欄第47行~第11欄第47行, 第9-10図 全文, 第1-24図 全文, 第1-24図	1, 2, 17 8-10 3-7, 11-16
A	JP, 7-116343, A (株式会社セガ・エンタープライゼス) 9. 5月. 1995 (09. 05. 95) (ファミリーなし) 全文, 第1-6図	1-17
<input checked="" type="checkbox"/> C欄の続きにも文献が列挙されている。 <input type="checkbox"/> パテントファミリーに関する別紙を参照。		
* 引用文献のカテゴリー 「A」 特に関連のある文献ではなく、一般的技術水準を示すもの 「E」 先行文献ではあるが、国際出願日以後に公表されたもの 「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す) 「O」 口頭による開示、使用、展示等に言及する文献 「P」 国際出願日前で、かつ優先権の主張の基礎となる出願日の後に公表された文献 「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの 「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの 「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの 「&」 同一パテントファミリー文献		
国際調査を完了した日 02. 09. 97		国際調査報告の発送日 09.09.97
国際調査機関の名称及びあて先 日本国特許庁 (ISA/JP) 郵便番号100 東京都千代田区霞が関三丁目4番3号		特許庁審査官 (権限のある職員) 植野 孝郎 電話番号 03-3581-1101 内線 3238



C (続き). 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
A	JP, 6-277362, A (株式会社ナムコ) 4. 10月. 1994 (04. 10. 94) (ファミリーなし) 全文, 第1-29図	7
Y	JP, 7-328228, A (株式会社セガ・エンタープライゼス) 19. 12月. 1995 (19. 12. 95) (ファミリーなし) 全文, 第1-6図	8-10
Y	JP, 7-191662, A (コナミ株式会社) 28. 7月. 1995 (28. 07. 95) (ファミリーなし) 全文, 第1-5図	8-10
A	JP, 8-137378, A (日立電子株式会社) 31. 5月. 1996 (31. 05. 96) (ファミリーなし) 全文, 第1図	11-12
A	JP, 7-105401, A (富士通株式会社) 21. 4月. 1995 (21. 04. 95) (ファミリーなし) 全文, 第1-18図	14-15
P, A	JP, 8-202342, A (株式会社カプコン) 9. 8月. 1996 (09. 08. 96) (ファミリーなし) 全文, 第1-7図	16